Escola Politècnica Superior
de Castelldefels
UNIVERSITAT POLITÈCNICA DE CATALUNYA

TRABAJO FINAL DE CARRERA

TÍTULO DEL TFC: Seguridad en WLAN 802.11: Evaluación de las contramedidas para combatir el ataque de falsificación

TITULACIÓN: Ingeniería Técnica de Telecomunicaciones, especialidad Telemática

AUTOR: Mireia García Dueñas

DIRECTOR: Elena López Aguilera

FECHA: 14-05-2010.

Título: Seguridad en WLAN 802.11: Evaluación de las contramedidas para combatir el ataque de falsificación

Autor: Mireia García Dueñas

Director: Elena López Aguilera

Fecha: 14-05-2010

Resumen

La movilidad, el coste y la facilidad de despliegue, son unas de las muchas ventajas que han hecho que las redes inalámbricas hayan experimentado un incremento importante a lo largo de las últimas décadas con respecto a las redes cableadas.

Otro factor que ha influido en la elección de las redes inalámbricas ha sido la evolución y mejora de sus prestaciones. El hecho de que el ancho de banda que ofrecen las últimas tecnologías wifi como por ejemplo 802.11n, ya no represente una gran diferencia con las redes cableadas, hace que se puedan ofrecer todo tipo de servicios como streaming de video en tiempo real u otras aplicaciones multimedia, eliminando parte de las limitaciones e inconvenientes que presentaban anteriormente.

Sin embargo, como toda tecnología presenta una serie de vulnerabilidades, como por ejemplo la sensibilidad a las interferencias y los errores de propagación entre otras, la convierten en el blanco perfecto de ataques que comprometen la información transmitida

La integridad, la confidencialidad, la autenticación, el control de acceso, la disponibilidad y el no repudio, son los objetivos principales que todo sistema de seguridad ha de conseguir para garantizar una red segura. Para ello se emplean variedad de filtros, métodos criptográficos y protocolos de seguridad.

El objetivo de este trabajo es evaluar la influencia que producen las contramedidas que se emplean para combatir el ataque de falsificación sobre el throughput, la justicia y el consumo de las baterías. Se estudia, se implementa y se compara los protocolos de autenticación EAP-TLS, EAP-PEAP, EAP-TTLS y EAP-LEAP en redes IEEE 802.11, mediante el IEEE 802.1X-EAP. Se utilizará un simulador de escenarios inalámbricos 802.11 cuyos resultados se contrastarán con un modelo analítico.

Title: Security in WLAN IEEE 802.11: Evaluation of countermeasures to combat forgery attack

Author: Mireia García Dueñas

Director: Elena López Aguilera

Date: 14-05-2010

Overview

Mobility, cost and ease of deployment, are among the many advantages which have made wireless networks get a significant increase over recent decades compared to wired networks.

Another factor that has influenced the choice of wireless networks has been the evolution and improvement of their performance. The fact that the bandwidth offered by the latest wireless technologies such as 802.11n, does not represent a major difference from wired networks, makes it able to offer all types of services such as streaming real time video or other multimedia applications, eliminating some of the limitations and drawbacks presented above.

However, like all technology it has a number of vulnerabilities, such as sensitivity to interference and propagation errors among others, making it the perfect target for attacks that compromise the information transmitted.

The integrity, confidentiality, authentication, access control, availability and non-repudiation, are the main objectives that every security system has to achieve to ensure a secure network. To that end we use a series of filters, cryptographic methods and security protocols.

The aim of this study is to evaluate the influence produced by countermeasures that are used to combat forgery attacks on the throughput and battery consumption. The authentication protocols EAP-TLS, EAP-PEAP, EAP-TTLS and EAP-LEAP in IEEE 802.11 networks are studied, compared and implemented, using the IEEE 802.1X-EAP. 802.11 wireless simulator scenario is used whose results will be compared with one analytical model.

En primer lugar agradecer a mi directora, Elena López, se esfuerzo y confianza para llevar a cabo este proyecto, así como su labor y esfuerzo dedicado.

En segundo lugar, una especial dedicatoria a mi pareja Miguel, por haberme ayudado, apoyado y comprendido en todo momento, sin él no podría haber llegado hasta el final. También quiero agradecer a mi familia y mis amigos por el soporte y la compañía ofrecida, ya que han hecho que la realización de este proyecto sea más llevadera.

Por último agradecer a David González de i2Cat, su ayuda con el mantenimiento de los servidores utilizados, ya que durante meses ha estado ayudándome con el soporte de éstos.

ÍNDICE

INTRODUCCIÓN

La movilidad, el coste y la facilidad de despliegue, son unas de las muchas ventajas que han hecho que las redes inalámbricas hayan experimentado un incremento importante a lo largo de las últimas décadas con respecto a las redes cableadas.

Otro factor que ha influido en la elección de las redes inalámbricas ha sido la evolución y mejora de sus prestaciones. El hecho de que el ancho de banda que ofrecen las últimas tecnologías wifi como por ejemplo 802.11n, ya no represente una gran diferencia con las redes cableadas, hace que se puedan ofrecer todo tipo de servicios como streaming de video en tiempo real u otras aplicaciones multimedia, eliminando parte de las limitaciones e inconvenientes que presentaban anteriormente.

Sin embargo, como toda tecnología presenta una serie de vulnerabilidades, como por ejemplo la sensibilidad a las interferencias y los errores de propagación entre otras, la convierten en el blanco perfecto de ataques que comprometen la información transmitida

La integridad, la confidencialidad, la autenticación, el control de acceso, la disponibilidad y el no repudio, son los objetivos principales que todo sistema de seguridad ha de conseguir para garantizar una red segura. Para ello se emplean variedad de filtros, métodos criptográficos y protocolos de seguridad.

El objetivo de este trabajo es evaluar la influencia que producen las contramedidas que se emplean para combatir el ataque de falsificación sobre throughput y el consumo de las baterías. Se estudia, se implementa y se compara los protocolos de autenticación EAP-TLS, EAP-PEAP, EAP-TTLS y EAP-LEAP en redes IEEE 802.11, mediante el IEEE 802.1X-EAP. Se utilizará un simulador de escenarios inalámbricos 802.11 cuyos resultados se contrastarán con dos modelos analíticos.

Para alcanzar dicho objetivo, el TFC se ha estructurado en cinco capítulos que se detallan a de la siguiente forma:

En el primer capítulo se establecen unas bases teóricas para realizar este estudio, en especial el funcionamiento de las redes WLAN y su seguridad.

En el segundo capítulo realizamos una descripción del ataque de falsificación, explicando su funcionamiento y las contramedidas que se emplean para combatir este ataque.

En el tercer capítulo se resume los mecanismos y medios empleados para la realización de las pruebas.

En el cuarto capítulo se evalúan los resultados obtenidos analizando los efectos producidos por el ataque en el throughput, la justicia y el consumo de baterías durante el ataque.

CAPÍTULO 1. REDES WLAN

En este capítulo analizaremos la tecnología IEEE 802.11 para tecnologías de redes de área local inalámbricas. En su inicio, 802.11 incluye la capa 802.11 MAC y dos capas físicas, la capa FHSS (Frequency hopping spread-spectrum) y la DSSS (Direct-sequence spread-sprectum), pero cada nueva especificación ha ido añadiendo nuevas características como veremos en este capítulo.

Además estudiaremos los tipos de redes inalámbricas existentes y los problemas a los que se enfrentan.

1.1 Nomenclatura y diseño

Las redes 802.11 constan de cuatro componentes principales tal como podemos observar en la figura 1.1.

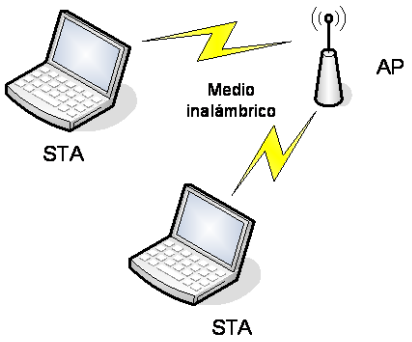


Fig. 1.1 Componentes de una red WLAN.

- **Estaciones (STA):** Elemento final en una red inalámbrica, por ejemplo, un ordenador portátil o de sobremesa con interfaz de red inalámbrica.

- **Access point (AP):** Componente que actúa como puente entre la red cableada y el medio inalámbrico, dando servicio a las estaciones.

- **Medio inalámbrico:** Compuesto de diferentes capas que permiten soportar 802.11 MAC, por ejemplo, la capa física de radiofrecuencia RF o la infrarroja, de las cuales la primera se ha estandarizado.

- **Sistemas distribuidos:** Conjunto de Access points que cubren un área comunicándose entre ellos para dar servicio a las estaciones móviles.

1.2. Tipos de redes inalámbricas

A continuación describiremos los tipos de redes existentes según su infraestructura:

- **Ad –Hoc:** También conocidas como IBSS (Set de Servicios Básicos Independiente). Se caracterizan por no existir ningún tipo de infraestructura o AP y las estaciones se comunican directamente entre sí (peer-to- peer). Limitadas al alcance de la interfaz de red inalámbrica de las estaciones. Se muestra en la figura 1.2:

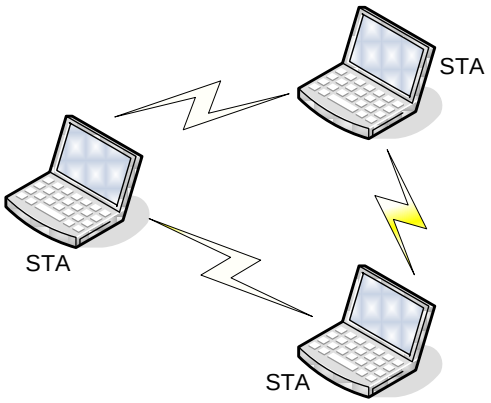


Fig. 1.2 Ad-Hoc

- **Infraestructura:** Da cobertura a un conjunto de dispositivos que deberán previamente asociarse al AP y accederán a la red cableada a través del él. Según su topología pueden ser BSS(Basic Service Set) donde encontramos una única celda que se comunica limitada por el alcance de su cobertura y la ESS (Extended Service Set), es la superposición de BSSs, cada una con su AP, que mediante un DS conectará todos los APs (véase figura 1.3). El cliente se moverá de BSS dentro del ESS de forma transparente al exterior.

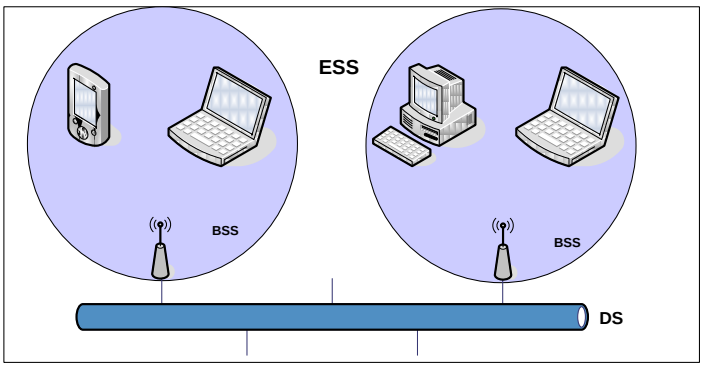


Fig. 1.3 Infraestructura

Esta es la infraestructura más empleada como sustituto de las redes cableadas convencionales. En este tipo de red los AP se comportan como hubs o switchs.

- **Híbridas:** Son una mezcla de las redes Ad-hoc y las de infraestructura.

1.3. Especificaciones 802.11

En 1997 se publicó el primer estándar IEEE para WLAN denominado 802.11. Sus tasas de transmisión son de 1 y 2 Mbps y opera en la banda de 2,4 GHz. Soporta como medio de transmisión infrarrojos y radiofrecuencia.

En la siguiente tabla podemos observar la evolución del estándar a día de hoy:

Tabla 1.1. Resumen de las características de los diferentes estándar 802.11.

Características	802.11	802.11b	802.11a	802.11g	802.11n
Año de estandarización	1997	1999	1999	2003	2010
Tasa de transferencia en capa física	2Mbps	11Mbps	54Mbps	54Mbps	600Mbps
Banda de frecuencia	2,4GHz	2,4GHz	5GHz	2,4 GHz	2,4GHz/5GHZ
Selección de frecuencia	FHSS / DSSS / IR	DSSS	OFDM	DSSS(OFDM)	DSSS(OFDM)

1.4. Acceso al medio

En este apartado nos centraremos en las redes con infraestructura. Este modo de operación puede funcionar bajo tres mecanismos: DCF(Distributed Coordination Function), PCF (Polling Coordination Function) y HCF (Hybrid Coordination Function).

1.4.1. DCF (Distributed Coordination Function)

Su acceso está basado en un mecanismo de contienda con CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) con backoff aleatorio y sus transmisiones son asíncronas.

1.4.1.1. CSMA/CA

En la figura 1.4 podremos ver el funcionamiento de CSMA /CA.

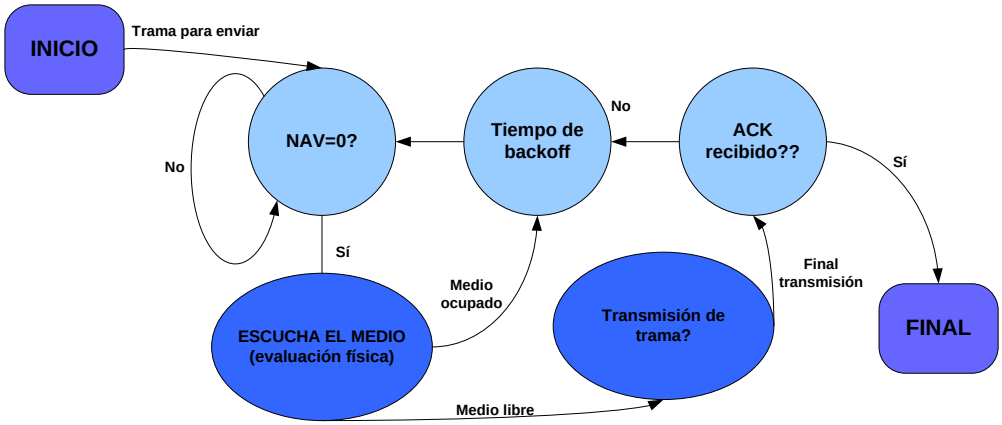


Fig.1.4 Mecanismo CSMA/CA

Este mecanismo se basa en la escucha o detección del medio antes de transmitir. El vector de ubicación de red (network allocation vector, NAV), es un contador residente en cada estación que representa la cantidad de tiempo que tardó en transmitirse la anterior trama de cualquier estación. El valor de NAV tiene que ser cero antes de que una estación intente enviar una trama. Si detecta que el medio está ocupado, por medio del mecanismo CCA (Clear Channel Assessment), su transmisión se retrasa hasta un momento posterior mediante el tiempo de backoff que describiremos en el siguiente apartado.

CCA consiste en la detección de actividad en el canal ya sea mediante la detección de una secuencia de bits concretas o un cierto de nivel de potencia por encima del umbral.

Si no se confirmase la transmisión mediante la recepción de un ACK se esperaría un tiempo de backoff antes de retransmitir el mensaje.

1.4.1.2. Algoritmo Backoff

Se invoca backoff cuando se detecta el medio ocupado antes de la transmisión, después de la transmisión y después de cada transmisión correcta. El comportamiento del tiempo de backoff tiene un valor exponencial cuyo valor se elige de forma aleatoria en el intervalo (0, W-1). W es la ventana de contención y su valor depende del número de intentos de transmisión fallidos. La primera vez que falla la transmisión de un paquete, la ventana de contención toma su valor mínimo CWmin, y por cada nueva transmisión fallida W dobla su valor anterior, hasta llegar a un límite $Wmax=2^{m}CWmin$. Los valores CWmin y CWmax, están estandarizados y su valor depende de la capa física utilizada, en la tabla 1.2 podemos ver dichos valores. Si el medio está libre el tiempo de backoff se decrementa, en cambio, si el medio vuelve a estar ocupado por la transmisión de otra estación, este decremento se frenará hasta que el medio vuelva a estar libre para así volver a restablecerse.

Tabla 1.2. Valores de CWmin y CWmax

PHY	CWmin	CWmax
FHSS	16	1024
DSSS	32	1024
IR	64	1024

IEEE 802.11e ofrece la posibilidad de escoger ventanas más pequeñas para el tráfico con prioridad más alta.

Como CSMA/CA no ofrece detección de colisiones, para poder dar diferentes prioridades, CSMA/CA define diferentes intervalos de separación entre tramas (IFS) (véase figura 1.5):

- **SIFS (Short Interframe Space):** Intervalo de tiempo más corto que se utiliza para transmitir información de control. Se emplea cuando se recibe un paquete con éxito y la estación que lo recibe espera un tiempo SIFS para enviar el ACK.

- **DIFS**: Intervalos de tiempo que se emplea para escuchar el canal antes de transmitir.

- **EIFS (Extended InterFrame space):** Es el tiempo esperado tras recibir una trama errónea que no puede entender.

Resaltar que el tiempo SIFS es inferior al tiempo DIFS para así priorizar los paquetes de reconocimiento respecto a los de datos.

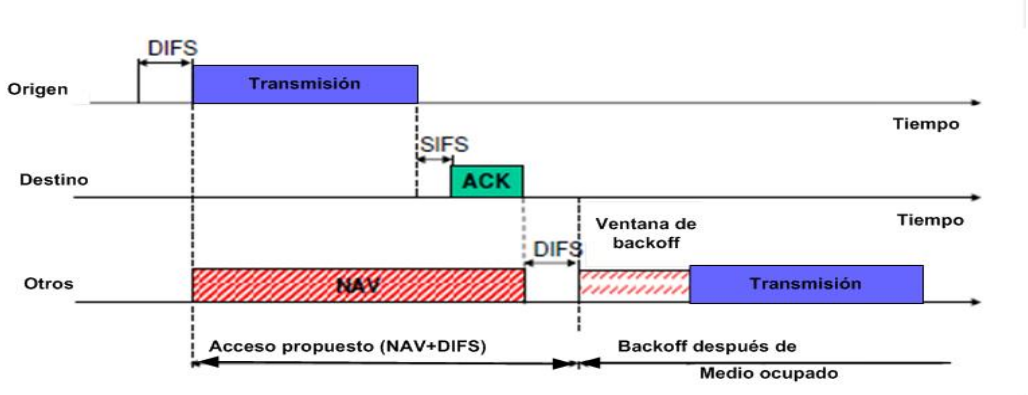


Fig. 1.5. Esquema de los diferentes intervalos de tramas (IFS).

1.4.2. PCF (Point Coordination Function)

El AP es el coordinador central y tiene una lista de las estaciones (polling list) que han solicitado transmitir. El tiempo de acceso se divide en Beacon Intervals, son intervalos periódicos cuyos valores los transmite el AP en TBTT (Target Beacon Transmission Time).

De la misma manera que tenemos en DFC los intervalos que determinan el acceso al medio DIFS y SIFS, en PCF encontramos el intervalo PIFS (PCF interframe space). PIFS sirve para que las estaciones que tienen datos para transmitir durante el periodo libre de contienda transmitan al acabar el periodo PIFS, evitando el tráfico basado en contienda.

1.5. Seguridad en 802.11 y sus problemas asociados

La integridad, la confidencialidad, la autenticación, el control de acceso, la disponibilidad y el no repudio, son los objetivos principales que todo sistema de seguridad ha de conseguir para garantizar una red segura. Para ello se emplean variedad de filtros, métodos criptográficos y protocolos de seguridad.

En este apartado vamos a analizar las distintas opciones de seguridad que han nacido del fruto de la necesidad de defenderse de los ataques que han ido surgiendo para explotar los problemas y vulnerabilidades del medio inalámbrico.

La tecnología 802.11 define dos sistemas para la securización: los sistemas pre-RSNA(Robust Security Network Association) y los RSNA. A continuación explicaremos los diferentes mecanismos de cifrado y autenticación que emplean cada uno de los sistemas anteriores además de las posibles soluciones que nos ayuden a responder ante los diferentes ataques.

1.5.1. Pre-RSNA

En 1999 el estándar 802.11 establece como protocolo de seguridad WEP (Wired Equivalent Privacy). El objetivo de WEP era proveer privacidad con el mismo nivel que ofrecían las redes cableadas, para ello ofrecía los servicios de privacidad e integridad de datos y autentificación de usuarios.

1.5.1.1. Autenticación

802.11 instauraba dos tipos de autenticación, la abierta (Open System Authentication) y de secreto compartido (PSK).

Como vemos en la Figura 1.9, la autenticación abierta no requiere ningún protocolo de autenticación pero sí existe un intercambio de mensajes donde la estación envía una solicitud de autenticación enviando su ID (Authentication

Request), el proceso se completa con una respuesta por parte del AP indicando su éxito o fracaso (Authentication Response).

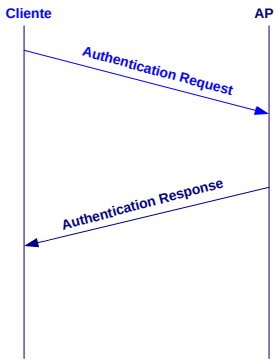


Fig.**1.9** Autenticación abierta

En el caso de la autenticación mediante el mecanismo de secreto compartido se requiere que el cliente configure una clave WEP. Como vemos en la figura 1.10, la estación envía una solicitud de autenticación, seguidamente el AP generará un desafió de 128 bits que enviará para que la estación cifre con la clave compartida mediante WEP. Una vez la estación cifre el desafío, lo copiará y lo enviará al AP donde éste lo descifrará y comparará con el que previamente le había enviado. Dependiendo de si los desafíos coinciden o no su respuesta será de éxito o fracaso. Este proceso es también denominado 4-Way-Handshake.

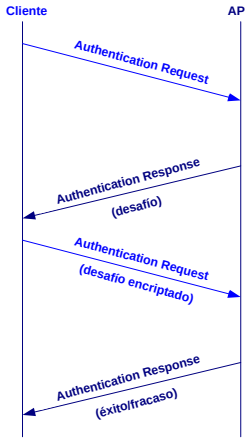


Fig. 1.10 Autenticación mediante mecanismo de secreto compartido

A pesar de no incluirse en el estándar, existe una tercer tipo de autenticación denominanada *MAC authentication*, en este caso, una vez el cliente envía su dirección *MAC*, el *AP* la compara con una lista de *MAC*'s a las cuales les está permitida la autenticación.

1.5.1.2. Cifrado e Integridad

Para cifrar WEP se utiliza RC4. Como se observa en la figura 1.11, mediante una semilla que está compuesta por 24 bits del vector IV(Initalization Vector) y 40 (o 104) bits de la clave compartida, el cifrador generará una clave de flujo pseudo-aleatoria que será sumada a los datos (XOR) para producir el mensaje cifrado. Además se le añade un campo de verificación de integridad ICV(Integrity Check value). Tanto ICV como los datos serán cifrados con la clave de flujo.

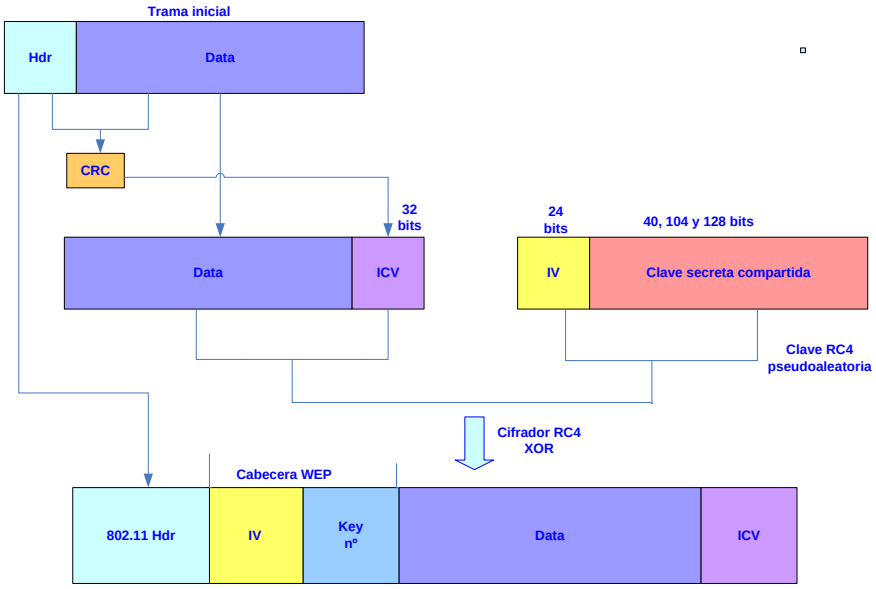


Figura 1.11. Creación de la trama WEP.

Una vez recibida la trama, el receptor obtendrá la misma clave de flujo utilizada en la encriptación añadiendo su clave compartida al vector IV extraído de la trama recibida. Una vez obtenida la clave de flujo se sumará (XOR) a los datos encriptados para así poder recuperar los datos y el ICV originales.

1.5.2. RSNA

Los sistemas RSNA proporcionan autenticación mediante la implementación de la IEEE 802.1X y EAP (Extensible Authentication Protocol). La confidencialidad e integridad de datos mediante TKIP y CCMP(CTR with CBC-MAC Protocol).

1.5.2.1. Autenticación

El proceso de autenticación y asociación empieza cuando una estación cliente selecciona un punto de acceso que anuncia su SSID (Service Set identifier). Seguidamente, la estación se asociará al AP elegido por defecto mediante la autenticación abierta. Una vez asociada la estación, empieza el proceso de

autenticación 802.11X y EAP. Para finalizar, se efectúa el proceso 4-Way-Handshake donde se negocian las claves criptográficas que se emplearán durante la transferencia de datos.

1.5.2.1.1. IEEE802.1X

IEEE 802.1X (véase figura 1.12) es un estándar del IEEE de control de acceso basado en puertos, este define tres entidades:

- **El solicitante o STA**: Es el usuario que quiere ser autenticado.
- **El autenticador (NAS):** Es el punto de acceso, controla el acceso físico a la red basado en el estado de la autenticación del cliente.
- **El servidor de autenticación (AS)**: es una entidad separada situada en la zona cableada (red clásica), pero también puede formar parte del punto de acceso.

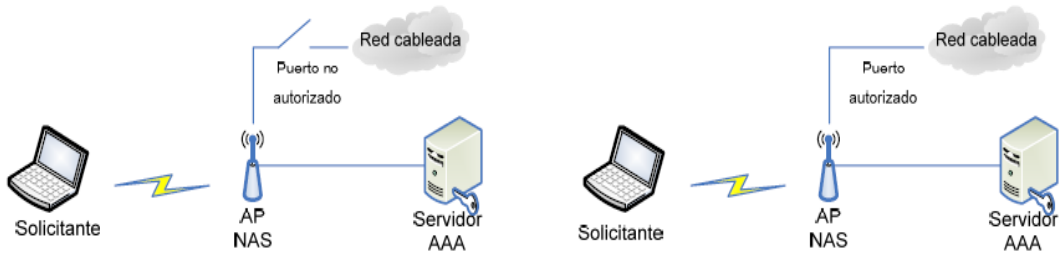


Fig.1.12 IEEE 802.1X

La estación solicitante y el autenticador intercambian información a través de un puerto descontrolado. El puerto controlado se encuentra bloqueado para el tráfico de datos hasta que el proceso de autenticación se completa correctamente sobre el puerto descontrolado. Así, mediante IEEE 802.1x se autentica en un puerto que no se utiliza para posteriores conexiones. La estación solicitante no puede transmitir datos hasta que se haya completado el proceso de autenticación.

1.5.2.1.2. EAP

El transporte de los mensajes de autenticación en 802.1X se realiza mediante EAP. Existen diferentes tipos:

- **EAP-MD5**: Es el método de autenticación EAP más antiguo y simple. No proporciona autenticación por parte del servidor. La autenticación de la

estación solicitante se realiza mediante la verificación de la función MD5 Hash de la contraseña de usuario. Este método actualmente no se utiliza ya que no es seguro.

- **EAP-TLS (transport Layer Security):** (véase [15]).Requiere de instalación de certificados en los clientes y en el servidor. Proporciona autenticación mutua fuerte (es decir, el servidor autentica al cliente y viceversa). La sesión de autenticación entre el cliente y el autenticador se cifra empleando el protocolo TLS (Seguridad en la Capa de Transporte).

- **EAP-TTLS (Tunneled TLS):** EAP-TTLS (véase [17]) es una extensión de EAP-TLS, fue desarrollado por Funk y Certicom. La autenticación EAP-TTLS usa certificados para autenticar el lado de la red. Por otro lado, emplea una manera menos compleja de autenticar el lado del solicitante, eliminando de esta manera la necesidad de configurar certificados en cada cliente WLAN. Establece un túnel seguro TLS para autenticar al cliente mediante otros protocolos de autenticación como PAP (Password Authentication Protocol), CHAP (Challenge Handshake Authentication Protocol), MSCHAP (Microsoft CHAP) o MSCHAPV2 (Microsoft CHAPv2).

- **PEAP:** (véase [16]).Protocolo propietario creado por Microsoft, Cisco y RSA Security. Funciona de manera parecida a EAP-TTLS, en el sentido de que solamente requiere del certificado de seguridad en el servidor. Se establece un túnel TLS a través del cual se procede a la utenticación del cliente con MSCHAPv2.

- **LEAP (Lightweight EAP)**: (véase [10]). Versión propietaria de Cisco basada en EAP-MD5. Autentica servidor y cliente mediante un secreto compartido (PSK) usando MSCHAPv2.

1.5.2.2. Confidencialidad e integridad de datos

El estándar 802.11i define dos algoritmos para la confidencialidad e integridad de datos: TKIP y CCMP. El primero es opcional pero el segundo es obligatorio para todos los sistemas RSNA.

1.5.2.2.1. TKIP

Es un mecanismo que mejora el mecanismo WEP. Al utilizar el mismo método de cifrado, el algoritmo RC4, puede operar en el hardware ya existente de WEP.

Para la integridad de datos, utiliza el algoritmo *Michael* para el cálculo de un MIC (Message Integrity Code). El MIC tiene un tamaño de 8 bytes y se coloca seguido de los bits de datos. MIC trabaja a nivel MSDU (MAC service Data Unit), por lo que si se produce fragmentación, no todos las MPDUs contienen el campo MIC.

Seguidamente, TKIP aplica WEP en cada MPDU con la diferencia de la clave RC4 que pasa a tener un tamaño de 128 bits y un IV de 48 bits. Estos dos últimos junto con el TSC (TKIP Sequence Counter), forman la semilla WEP que es diferente para cada MPDU.

Como vemos en la figura 1.13, TKIP expande en 20 bytes el tamaño de la MPDU original, 4 bytes para el campo IV, 4 bytes para la extensión IV, 4 bytes para el ICV y 8 bytes para el MIC.

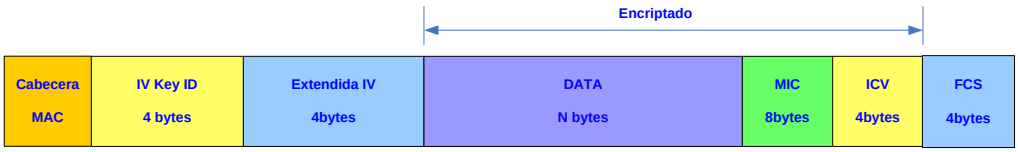


Fig.1.13 trama TKIP

TKIP ha sido diseñado para ser robusto contra ataques de falsificación y replay. Cuando TKIP está sufriendo un ataque de falsificación aplica una serie de contramedidas que especificaremos en el apartado 2.1.

1.5.2.2.2. CCMP

Este mecanismo proporciona confidencialidad y autenticación además de integridad de datos y de cabecera MPDU. Sustituye el algoritmo RC4 y emplea AES (Advance Encryption Standard).

Para construir el MIC y generar el texto cifrado emplea una clave AES de 128 bits. Para ello se añade al algoritmo un PN (Packet Number) que se incrementará y será diferente para cada MPDU.

A diferencia de TKIP, CCMP incrementa en 16 bytes el formato de la MPDU, 8 bytes de cabecera CCMP y 8 bytes para el MIC (véase figura 1.14).

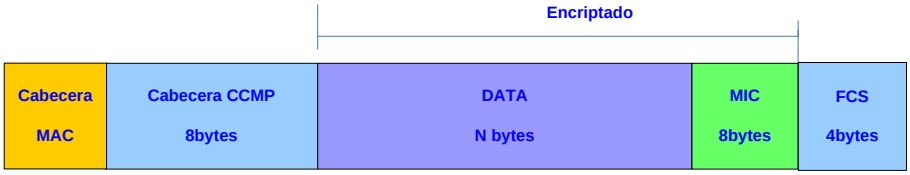


Fig. **1.14** Trama CCMP.

El resultado de combinar el mecanismo 802.1X-EAP para la autenticación y CCMP para la confidencialidad e integridad de datos, resulta un esquema mucho más complejo que el inicial WEP. Aunque la complejidad computacional y los problemas de interoperabilidad aumenten, la seguridad que proporciona este esquema compensa estos inconvenientes.

1.5.2.3. WPA y WPA2

WPA (Wi-Fi Protected Access) fue diseñado para sustituir a corto plazo a WEP. Emplea para la autenticación el sistema 802.1X –EAP y para la integridad y confidencialidad de datos TKIP.

WPA2 consiste en la ratificación de la Wi-Fi Alliance del estándar IEEE 802.11i, donde utiliza 802.1x para la autenticación y TKIP y CCMP para la confidencialidad e integridad de datos.

CAPÍTULO 2. ATAQUE DE FALSIFICACIÓN

Las contramedidas que proporciona TKIP para proteger de los ataques de falsificación, puede provocar DoS.

En este capítulo estudiaremos cómo se genera el ataque de falsificación para así poder analizar en capítulos posteriores sus consecuencias.

2.1 Contramedidas TKIP

Uno de los fallos más significativos de WEP fue la falta de mecanismo para combatir la falsificación de mensajes y otros tipos de ataques similares, como por ejemplo el ataque de cambio de bits, modificación de datos mediante concatenación y truncamiento, etc.

MIC dificulta estos ataques pero no elimina la posibilidad de que sucedan. TKIP cifra el MIC para dificultar su falsificación y proporciona la detección de repeticiones mediante la secuencia del TSC y la validación del ICV.

Cuando se produce un fallo en la validación del MIC, TKIP como medida de seguridad ejecuta unas contramedidas para lograr dos objetivos:

- Los eventos que producen errores en el MIC deben ser considerados como asuntos relevantes de seguridad ya que un error en la MIC generalmente es un indicador de un ataque activo.

- El número de errores en la MIC debe mantenerse por debajo de dos por minuto. Si se detectan dos errores en menos de 60s se deben deshabilitar todas las recepciones por un periodo de 60s. Este retardo dificulta los ataques de muchos intentos en poco tiempo (ataques iterativos para adivinar contraseñas como los ataques de fuerza bruta).

Antes de comprobar el MIC, el receptor debe verificar el FCS, IVC y el TSC vinculados. Para evitar errores en el MIC innecesarios, cualquier trama que tenga su FCS, un ICV o un valor de TSC inválido, debe descartarse antes de comprobar el MIC.

El primer aviso de error del MIC debe ser registrado y un temporizador debe iniciarse para habilitar la ejecución de las contramedidas. Si se producen dos fallos de MIC durante un periodo de 60s, se procede a la desautenticación enviando una trama EAP request Failure.

Si es el AP quien detecta el segundo fallo en la comprobación del MIC, éste desautentica a todas las estaciones y elimina sus claves correspondientes. Pasado 60 s, el proceso de autenticación 802.1x – EAP puede iniciarse nuevamente.

Si el evento es detectado por la STA (suplicante), ésta debe ser reportada al AP enviando la trama "Michael MIC Failure Report" (véase anexo A). Inmediatamente, se desautenticará del AP y pasados 60s podrá empezar el proceso de autenticación 802.1X –EAP (véase figura 2.1)

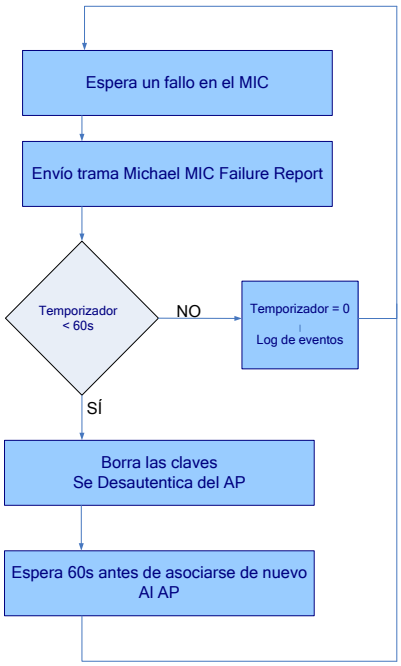


Fig. 2.1 Contramedida para una STA

2.2 Plan de ataque

Como hemos dicho anteriormente, las contramedidas que emplea TKIP para combatir el ataque de falsificación, puede provocar DoS.

Para llevar a cabo este ataque hemos analizado dos escenarios, el primero el AP el que detecta el último fallo del MIC y el segundo son las estaciones las que lo detectan.

2.3 Metodología

La metodología de este ataque consiste en engañar al AP o al STA (dependiendo del caso) haciéndoles creer que son atacados, de esta forma, forzamos a que el mecanismo TKIP emplee sus contramedidas de seguridad provocando la desautenticación forzada de los dispositivos.

Como consecuencia del ataque, los dispositivos que se ven afectados por esta desautenticación experimentarán una imposibilidad de transmitir que les afectará en parámetros como el throughput, la injusticia y el consumo. En próximos capítulos analizaremos en qué medida afecta el ataque en estos parámetros.

CAPÍTULO 3. HARDWARE Y SOFTWARE EMPLEADO

A continuación mostraremos el hardware y software utilizado para la realización de este trabajo.

3.1. Hardware utilizado

Dada la naturaleza de este estudio, sólo ha sido necesario emplear el hardware en el cual se ejecutaban las pruebas. Las características de estas estaciones se detallan en la siguiente tabla.

Tabla 3.1 Descripción del hardware empleado

Nombre	Estación 1	Estación 2	Estación 3
Marca	Sony	Dell	(clónico)
Modelo	VGN-S5M	Poweredge 2950	(clónico)
Memoria Ram	1GB	4.00 GB	3.50 GB
Procesador	Intel Pentium M740 1,73 GHz	Intel XEON E5335 2.00 GHz	Intel Pentium D 3.00 GHz
Sistema operativo	Windows XP SP3	Windows 2003 R2 server SP2	Windows 2003 server SP2
Disco duro	80 GB	120 GB	650 GB

3.2. Software utilizado

3.2.1. Escritorio remoto de Windows

Se utilizo esta aplicación de Microsoft para el control remoto de las estaciones 2 y 3. Esta aplicación es nativa de todos los sistemas operativos de Microsoft, en concreto, durante la realización de este trabajo se ejecutó la aplicación desde la estación 1 con sistema operativo Windows XP SP3.

3.2.2.Borland C++ Builder

Entorno de programación para C++ sobre el cual se ha programado el simulador 802.11 y se han realizado las modificaciones necesarias para efectuar las pruebas con el simulador. Este software pertenecía a la empresa Borland que fue adquirida por Embarcadero Technologies en 2009.

3.2.3.Simulador 802.11

Para analizar los efectos que nos produce el ataque de falsificación se ha utilizado como herramienta el simulador 802.11. Este simulador realizado en Borland C++ opera sobre plataformas Windows. En el anexo B encontraremos una manual donde se explica de forma detallada el funcionamiento de éste.

CAPÍTULO 4. PRUEBAS Y RESULTADOS

Mediante el simulador 802.11 se investigará la influencia que produce el ataque DoS sobre el throughput, la justicia y el consumo de las baterías. Se evaluarán estos factores con la implementación y la comparación entre los protocolos de autenticación EAP-TLS, EAP-PEAP, EAP-TTLS y EAP-LEAP en redes IEEE 802.11 mediante el IEEE 802.1X-EAP. Para llevar a cabo este estudio, se analizarán dos escenarios, el primero cuando el AP es atacado y el segundo cuando son las STA las atacadas.

Durante la investigación, se realizan pruebas con los diferentes protocolos de autenticación, pero se ha hecho especial hincapié en TLS ya que es uno de los más empleados.

4.1. Metodología

Para la realización de las simulaciones se ha elegido un escenario unicelular que tiene actuando un AP y múltiples STAs, todas ellas compartiendo un mismo medio. Mediante intervalos de backoff y de tiempos de espera DIFS y SIFS todas las estaciones puedan acceder al medio de forma equitativa y aleatoria. En la tabla 4.1, se muestran los valores de los parámetros que definen el acceso al medio considerados en las simulaciones.

Tabla 4.1. Parámetros del simulador 802.11

Tiempo de simulación	18000 s
Cabecera MAC	34 bytes
Velocidad de transmisión	54 Mbps
Tiempo de slot	9 µs
Tiempo DIFS	28µs
Tiempo SIFS	10 µs
ACK	14 bytes
Tráfico Ofrecido para tramas 600,1000 y 1500 bytes.	0,9 (Saturación)
Tráfico Ofrecido para tramas de 200 bytes.	(Saturación)
Ventana de backoff	CWmin=16 CWmax=1024

En este proyecto se evalúan los diferentes escenarios derivados del trabajo final de carrera de Didac Mediavilla [3]. La tablas 4.2 y 4.3 recogen las descripciones de los equipos portátiles y tarjetas empleadas en dichos escenarios respectivamente.

Tabla 4.2 Descripción de los equipos portátiles.

Nombre	Pc1	Pc2	Pc3
Marca	Acer	Acer	Acer
Modelo	Aspire 1690	Extensa 5620z	Extensa 5620z
Memoria RAM	512 Mb	2048Mb DDRII	2048Mb DDRII
Procesador	Pentium M 1.6GHz	Intel Pentium Core Duo T2310/ 1.46 GHz	Intel Pentium Core Duo
Sistema operativo	Windows XP Professional	Windows XP Professional	Windows Vista Home Premium
Capacidad de la batería [mAh]	4400	4000	4000
Duración de batería [h]	3	2,5	2,5
Tarjeta Wireless instalada	Intel PRO/Wireless 2200BG	Atheros AR5007	Atheros AR5007

Tabla 4.3 Tarjetas empleadas.

Marca	3Com	Intel	Atheros	Cisco
Modelo	3CRUSB10075	Intel PRO/Wireless 2200BG	Atheros AR5007	Cisco Aironet Wireless CardBus Adapter
Estándares WLAN	802.11b/g	802.11b/g	802.11b/g	802.11a/b/g
Canales	1 - 13	1 - 14	1 - 14	1 – 14
Seguridad	TLS, PEAP, TTLS	TLS, PEAP, TTLS y LEAP	TLS, PEAP, TTLS y LEAP	TLS, PEAP, TTLS y LEAP
Sistema operativo	Windows Xp, Vista y Linux	Windows XP y Linux	Windows Xp, Vista y Linux	Windows XP y Linux
Potencia Tx [dBm]	16	16	17	20
Consumo Idle [mW]	750	60	660	669,9
Comsumo Tx [mW]	1000	1450	1419	1828,2
Consumo Rx [mW]	1000	850	1419	1049,4
Software 3Com	OfficeConect Utility	Intel PRO/Wireless 2200BG	Atheros Client Utility	Aironet Desktop Utility
Chipset	Zydas ZD1211	Intel	AR2425	Atheros5002X

Para simular este ataque distinguiremos tres tiempos: el tiempo de simulación, el tiempo de ataque y el periodo de ataque, tal y como muestra la figura 4.1.

El tiempo de ataque es la duración de nuestro ataque y toma el valor de la suma de los 60 segundos (tiempo que tiene que esperar una STA para volver a autenticarse), del retardo de la desautenticación y el retardo de autenticación, el cual es medido a partir de los escenarios reales descritos anteriormente en las tablas 4.2 y 4.3 Estos valores los podemos ver de forma más detallada en el anexo C.

El periodo del ataque es el tiempo que marca cada cuánto tiempo ejecutaremos el ataque y el tiempo de simulación, es el tiempo que marcará la duración de la simulación.

En los escenarios analizados durante las pruebas, se han aplicado periodos de ataque con valores múltiplos del tiempo de ataque.

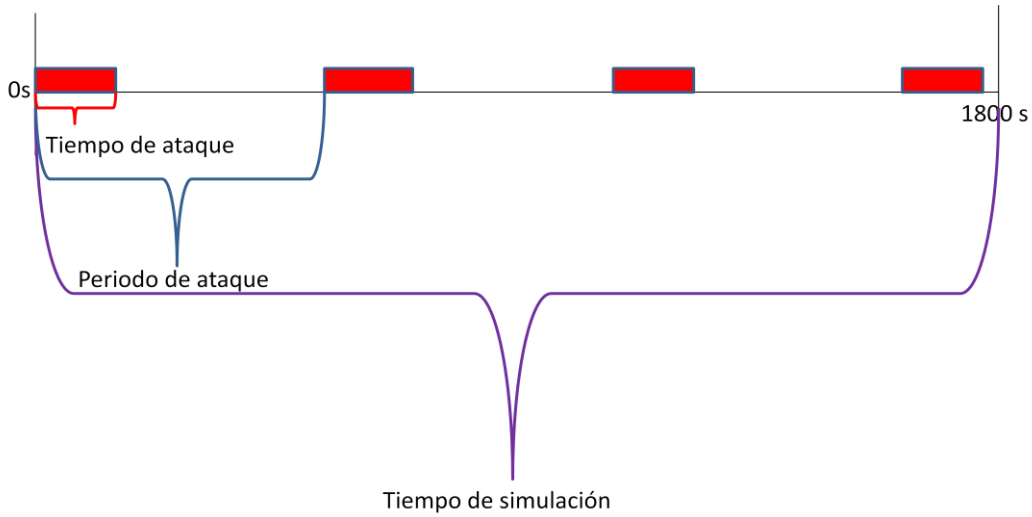


Fig. 4.1. Diagrama de tiempos.

4.1.1. Simulaciones para el escenario 1: El AP es el atacado

Con este escenario simularemos las contramedidas que utiliza TKIP cuando el AP está siendo atacado. Para ello, se configura el simulador de forma que el AP no puede transmitir durante el tiempo en el que éste está siendo atacado y las estaciones se tienen que volver a autenticar.

4.1.2. Simulaciones para el escenario 2: Las STAs son las atacadas

En el escenario 2 se estudia qué sucede cuando el protocolo TKIP pone en marcha la contramedida cuando una STA está siendo atacada.

Para simular este escenario se modifica parte del código del simulador 802.11. Con esta modificación, se elijen las estaciones que son atacadas y se

establece que para cada periodo de ataque, las STAs atacadas no puedan transmitir por un tiempo de ataque. En el anexo D vemos la parte de código que fue modificado

4.2. Estudio del throughput

4.2.1. El AP es atacado

En esta sección se estudiará cómo afecta en el throughput el ataque de falsificación para el escenario 1, donde el AP está siendo el atacado.

4.2.1.1. Influencia del número de STAs.

4.2.1.1.1. Plan de pruebas

El objetivo de esta prueba es analizar cómo influye el número STAs en el throughput. Para ver esta influencia se estudiarán escenarios con 2, 4,10, 15, 20, 50 y 100 STAs, enviando paquetes de 200, 600, 1000 y 1500 bytes.

La tabla 4.4 muestra los parámetros fijados para esta prueba.

Tabla 4.4 Valores fijados para la influencia del número de STAs.

Velocidad	54 Mbps
Tiempo de simulación	1800 s
Periodo de ataque	603,34 s [1]
Tráfico ofrecido (Normalizado)	0,9 (Saturación)
Protocolo de autenticación	EAP- TLS
Tarjeta y equipo empleado	·3Com Pc1

4.2.1.1.2. Resultados obtenidos

Como podemos observar en la figura 4.2, a medida que el número de STAs aumenta, el throughput decrementa, por el contrario, si se incrementa el tamaño de paquetes de datos, el throughput aumenta.

Asimismo, el punto más óptimo resulta ser un throughput de 0,56 con un paquete de datos de 1500 bytes y siendo 2 el número de STAs.

[1] Los periodos de ataque son múltiplos del tiempo de ataque y tomarán valor según el escenario analizado (véase anexo C). Consultando el anexo C, si contemplamos el caso para el protocolo EAP-TLS con la tarjeta ·3Com y el Pc1, el tiempo de ataque es de 60,334s. En este caso, el periodo de ataque que es 603,334s, es 10 veces el tiempo de ataque.

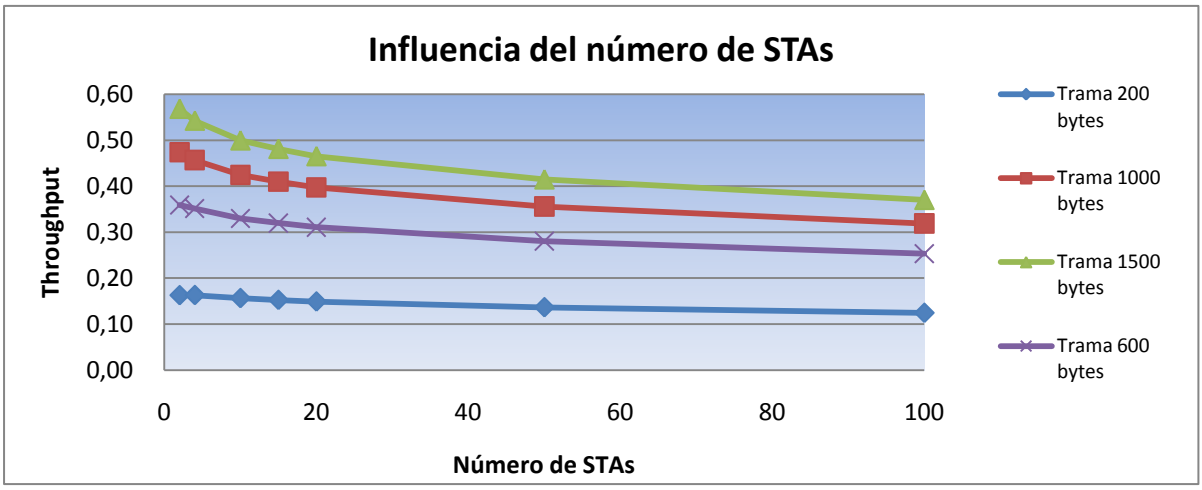


Fig.4.2 Influencia del número de STAs

4.2.1.2. Influencia del periodo de ataque

4.2.1.2.1. Plan de pruebas

En este punto se examina el impacto que produce la variación del periodo de ataque en el throughput. Se analizan escenarios con periodos de ataque de 120,668 s (2 veces el tiempo de ataque), 301,67 s (5 veces el tiempo de ataque), 603,34 s (10 veces el tiempo de ataque), 1206,68 s (20 veces el tiempo de ataque), 1800s (marca el umbral ya que no se produce ningún ataque) y 2800s (por encima del umbral), enviando paquetes de 200,600, 1000 y 1500 bytes.

En la tabla 4.5 se muestran los parámetros fijados para esta prueba.

Tabla 4.5 Valores fijados para la influencia del periodo de ataque.

Velocidad	54 Mbps
Tiempo de simulación	1800 s
Número de estaciones	10
Tráfico ofrecido (Normalizado)	0,9 (Saturación)
Protocolo de autenticación	EAP- TLS
Tarjeta y equipo empleado	·3Com Pc1

4.2.1.2.2. Resultados obtenidos

En la figura 4.3 se comprueba que a medida que el periodo de ataque aumenta el throughput aumenta. Esto se debe porque al ser más grande el periodo de repetición, menos ataques se producen durante la simulación y por lo tanto, más datos podrán transmitir las STAs.

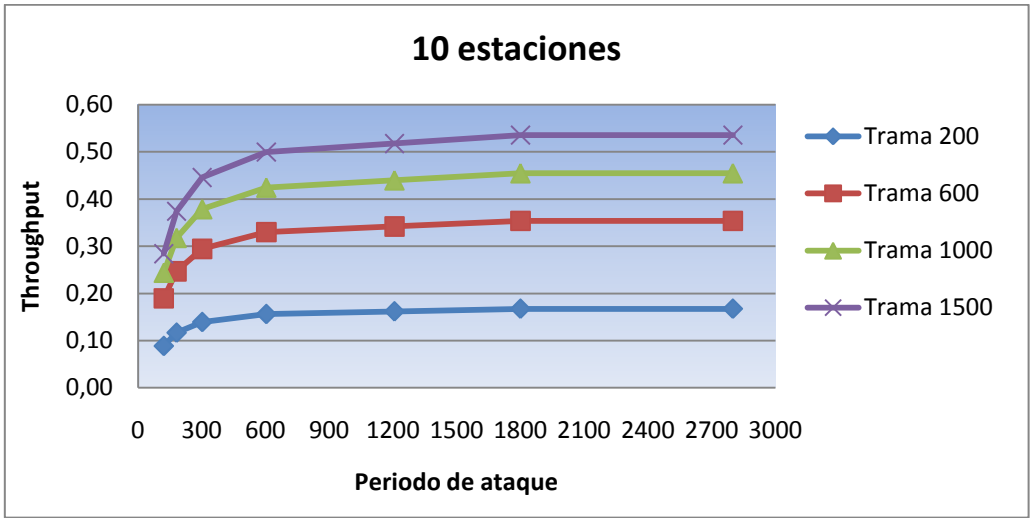


Fig. 4.3 Influencia del periodo de ataque

Por otra parte, resaltar que a partir de los 1800 s, si se aumenta el periodo de ataque, los resultados no variarán ya que los 1800s del tiempo de simulación marcan el umbral. De este modo los resultados de 1800 y 2800s son los mismos.

4.2.1.3. Influencia del número de usuarios variando el periodo de ataque

4.2.1.3.1. Plan de pruebas

En esta sección, el estudio se centra en la variación del periodo de ataque y el número de STAs enviando paquetes de 200, 600, 1000 y 1500 bytes.

Para llevar a cabo estas pruebas, a cada paquete de datos de 200, 600, 1000 y 1500 bytes se les aplica variaciones de 2, 4, 10, 15, 20, 50 y 100 STAs con periodos de ataque de 120,668 s, 301, 67 s, 603,34 s, 1206,68 s y 1800s.

En la tabla 4.6 se muestran los parámetros fijados para estas pruebas:

Tabla 4.6 Valores fijados para la influencia del número de STAs y el periodo de ataque

Velocidad	54 Mbps
Tiempo de simulación	1800 s
Tráfico ofrecido (Normalizado)	0,9 (Saturación)
Protocolo de autenticación	EAP- TLS
Tarjeta y equipo empleado	·3Com Pc1

4.2.1.3.2. Resultados obtenidos

Observando las figuras 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 4.10, podemos concluir que aumentando el periodo de ataque y el tamaño del paquete de datos, el throughput aumenta. De lo contrario, aumentando el número de estaciones el throughput disminuye.

La tabla 4.7 resume los resultados de los throughputs máximos conseguidos para cada tamaño de paquetes de datos con un periodo de ataque máximo de 1800 s y 2 STAs.

Tabla 4.7 Resumen de los throughputs máximos.

Tamaño del paquete	Throughput máximo
200 bytes	0,17
600 bytes	0,38
1000 bytes	0,51
1500 bytes	0,61

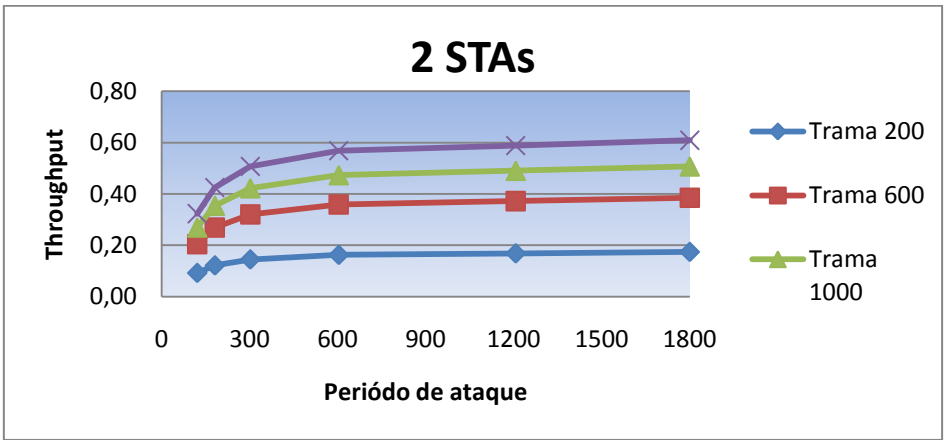


Fig. 4.4 Influencia del número de STAs y periodo de ataque para 2 STAs.

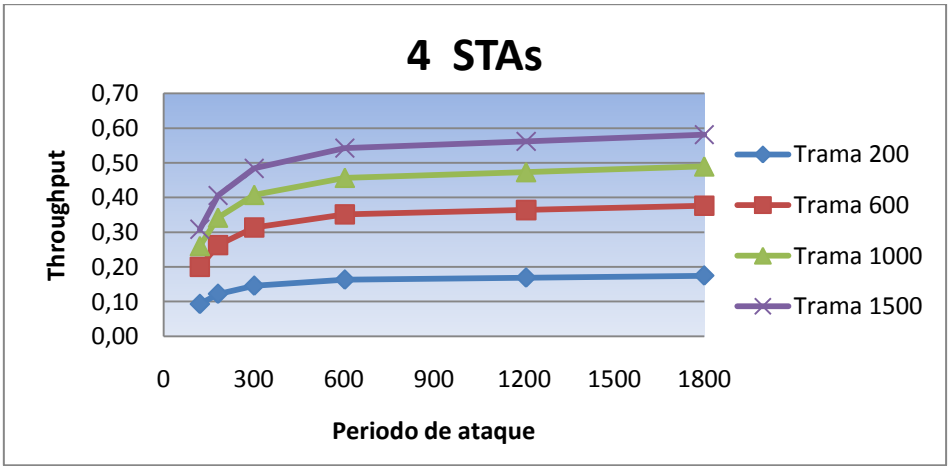


Fig. 4.5 Influencia del número de STAs y periodo de ataque para 4 STAs.

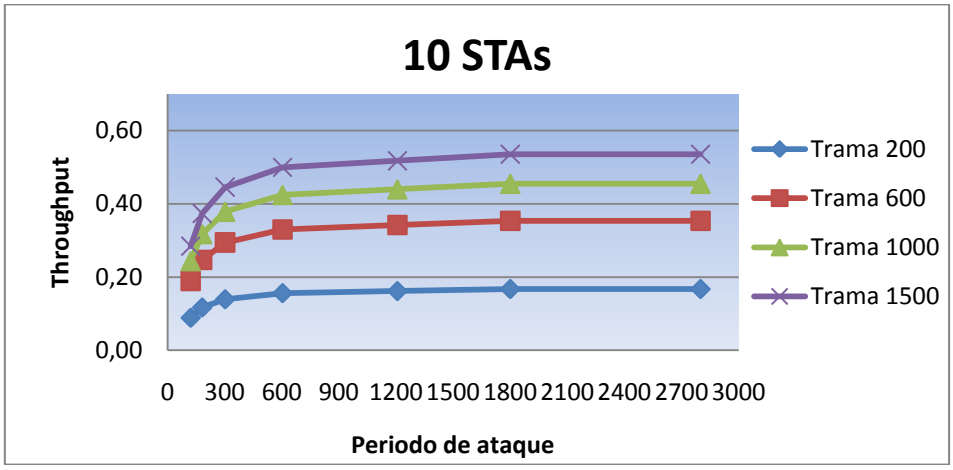


Fig. 4.6 Influencia del número de STAs y periodo de ataque para 10 STAs.

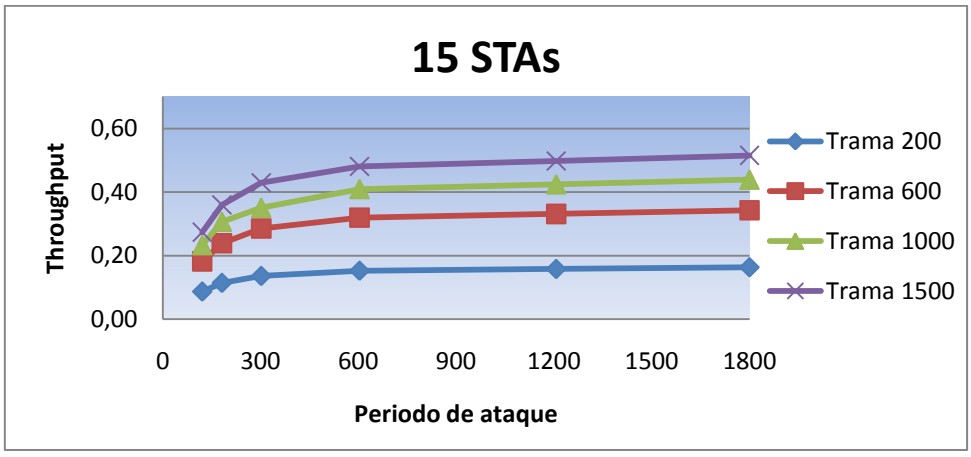


Fig. 4.7 Influencia del número de STAs y periodo de ataque para 15 STAs.

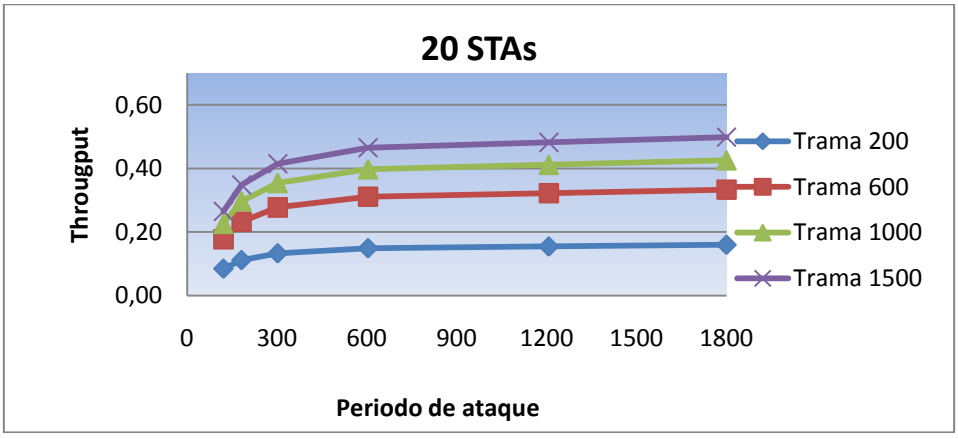


Fig. 4.8 Influencia del número de STAs y periodo de ataque para 20 STAs.

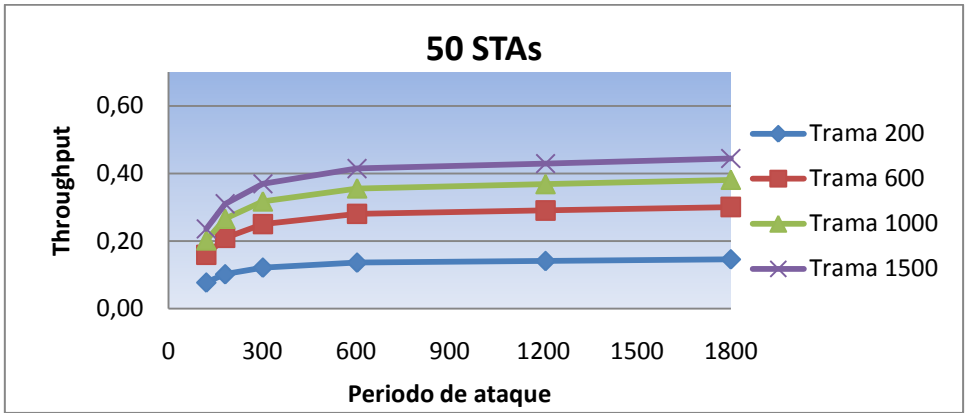


Fig. 4.9 Influencia del número de STAs y periodo de ataque para 50 STAs.

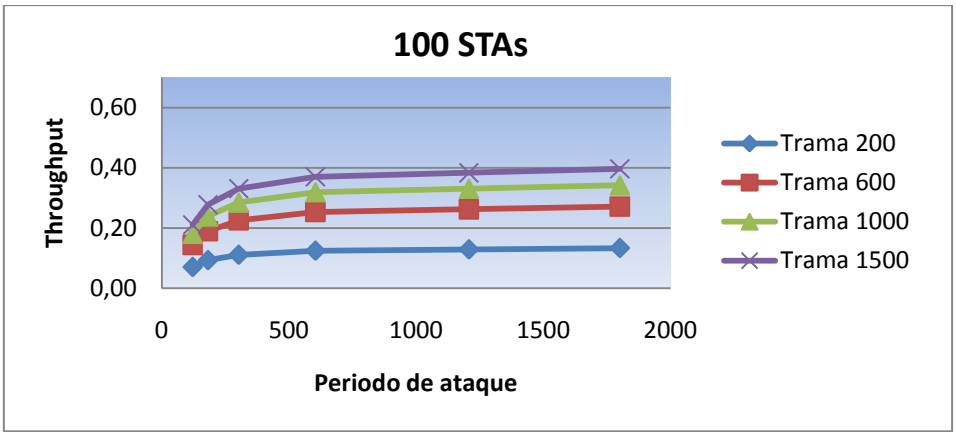


Fig. 4.10 Influencia del número de STAs y periodo de ataque para 100 STAs.

La tabla 4.8 resume los incrementos observados para el caso de 2 y de 100 estaciones, entre un periodo de 120,668 s y otro de 1800 s, siendo éstos los casos más extremos, extraídos de las figuras anteriores.

Tabla 4.8 Resumen de porcentaje de diferencia del throughput.

	2	100
200 bytes	46,87%	46,95%
600 bytes	46,90%	46,93%
1000 bytes	46,94%	46,97%
1500 bytes	46,92%	46,97%

Como se puede observar, la proporción entre los resultados de los distintos periodos para las tramas analizadas se mantiene constante, independientemente del número de estaciones y del tamaño de la trama.

4.2.1.4. Influencia de la variación del tráfico ofrecido en el throughput.

4.2.1.4.1. Plan de pruebas

Se pretende estudiar de qué manera influye el tráfico ofrecido en el throughput. Para ver esta influencia se variará el tráfico de 0,1 en 0,1 en paquetes de datos de 200, 600, 1000 y 1500 bytes.

En la tabla 4.9 se muestran los parámetros fijados para esta prueba.

Tabla 4.9 Parámetros fijados para la influencia del tráfico ofrecido.

Velocidad	54 Mbps
Tiempo de simulación	1800 s
Tamaño de datos	1500 bytes.
Protocolo de autenticación	EAP- TLS
Número de estaciones	10
Periodo de ataque	603,34 s

4.2.1.4.2. Resultados obtenidos

Como muestra la figura 4.11, a medida que el tráfico ofrecido aumenta el throughput también incrementa.

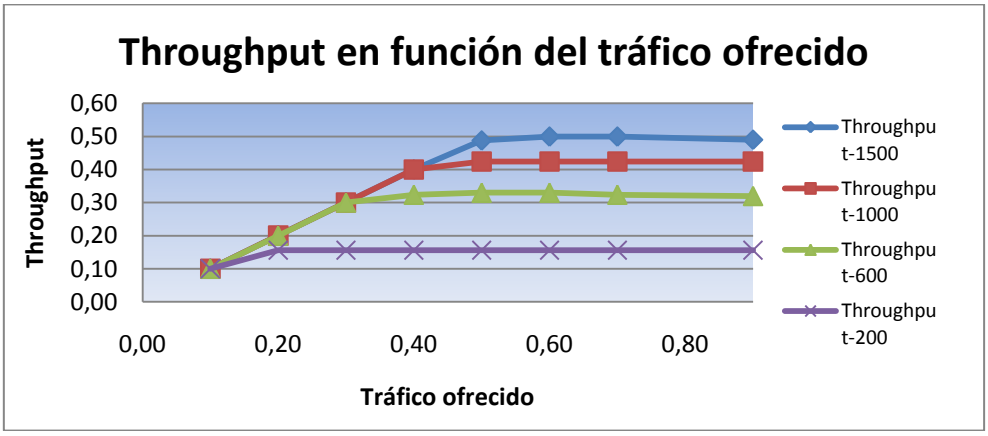


Fig.4.11 Influencia del tráfico ofrecido.

Además de ver la influencia que produce la variación del tráfico ofrecido, en la tabla 4.10 se puede observar como en el caso de 1500 bytes tarda más en saturarse. Esto se debe a que el tamaño de datos a enviar en un paquete es mayor y por lo tanto se cursa más tráfico ofrecido que cualquier paquete que transporte menor cantidad de datos.

Tabla 4.10 Resumen de los puntos en el que se satura el throughput.

Tamaño del paquete de datos	Punto de saturación del tráfico ofrecido
200 bytes	0,15
600 bytes	0,32
1000 bytes	0,42
1500 bytes	0,49

4.2.1.5. Estudio de los diferentes protocolos de autenticación para las diferentes escenarios.

4.2.1.5.1. Plan de pruebas.

En esta parte del estudio se pretende analizar qué protocolos de autenticación son los que tienen mejor respuesta frente al ataque DoS, considerando los diferentes escenarios mostrados en la tabla 4.11.

La tabla 4.11 recoge cada uno de los escenarios que se analizarán durante el proyecto. Estos escenarios son extraídos del proyecto final de carrera de Didac Mediavilla [3] y vienen determinados por el conjunto que forman la tarjeta y el equipo empleado (descritos anteriormente en las tablas 4.2 y 4.3).

Tabla 4.11 Escenarios utilizados en el estudio.

EAP-TLS	3Com-pc1, 3Com-pc2, 3Com-pc3, Intel-Pc1, Atheros-pc2, Atheros-pc3, Cisco-Pc1, Cisco-Pc2
EAP-PEAP	3Com-pc1, 3Com-pc2, 3Com-pc3, Intel-Pc1, Atheros-pc2, Atheros-pc3, Cisco-Pc1, Cisco-Pc2
EAP-TLS	3Com-pc1, 3Com-pc2, 3Com-pc3, Intel-Pc1, Atheros-pc2, Atheros-pc3, Cisco-Pc1, Cisco-Pc2
EAP-LEAP	Intel-Pc1, Atheros-Pc2, Cisco-Pc1, Cisco-Pc2
EAP-LEAP (AP+RADIUS) En este escenario el autenticador es el servidor RADIUS	Intel-Pc1, Atheros-Pc2, Cisco-Pc1, Cisco-Pc2
EAP-TTLS (software Intel)	Intel-Pc1 con los métodos PAP, CHAP, MSCHAP, MSCHAPv2.

Para analizarlo se aplicarán todos los protocolos en cada una de los escenarios variando los periodos de ataque con valores de 120,668 s, 301,67s, 603,34 s, 1206,68s y 1800s. Estos valores variarán según el escenario escogido.

Según el escenario, el AP estará más tiempo o no sin transmitir dependiendo de la latencia del proceso de autenticación de cada escenario. La tabla 4.12 muestra los retardos de autenticación para cada uno de los escenarios. Estos valores son extraídos del proyecto final de carrera de Dídac Mediavilla [3].

Tabla 4.12 Retardos de autenticación.

Tiempo [s]	3Com pc1	3Com pc2	3Com pc3	Intel pc1	Atheros pc2	Atheros pc3	Cisco pc1	Cisco pc2
EAP-TLS	0,333	0,222	0,171	0,291	0,190	0,289	0,288	0,199
EAP-PEAP	0,286	0,282	0,177	0,250	0,176	0,309	0,273	0,178
EAP-TTLS	0,817	0,520	0,530	0,690	0,579	0,728	0,619	0,615
EAP-LEAP	-	-	-	0,142	0,066	-	0,193	0,070
EAP-LEAP (AP+Radius)	-	-	-	0,080	0,058	-	0,151	0,060
EAP-TTLS (software Intel)	-	-	-	0,141	-	-	-	-

La tabla 4.13 presenta los valores fijados para esta sección de pruebas.

Tabla 4.13 Parámetros fijados para la comparativa de las diferentes tarjetas con los distintos protocolos de autenticación.

Velocidad	54 Mbps
Tiempo de simulación	1800 s
Tamaño del paquete de datos	1500 bytes
Número de STAs	10

4.2.1.5.2. Resultados obtenidos.

Como vemos en las figuras 4.12 - 4.19, el protocolo de autenticación que mejor responde frente al ataque es el que menor tiempo tarda en completar con éxito el tiempo de autenticación. Dicho protocolo de autenticación, obtendrá un throughput más elevado a medida que el periodo de ataque sea mayor.

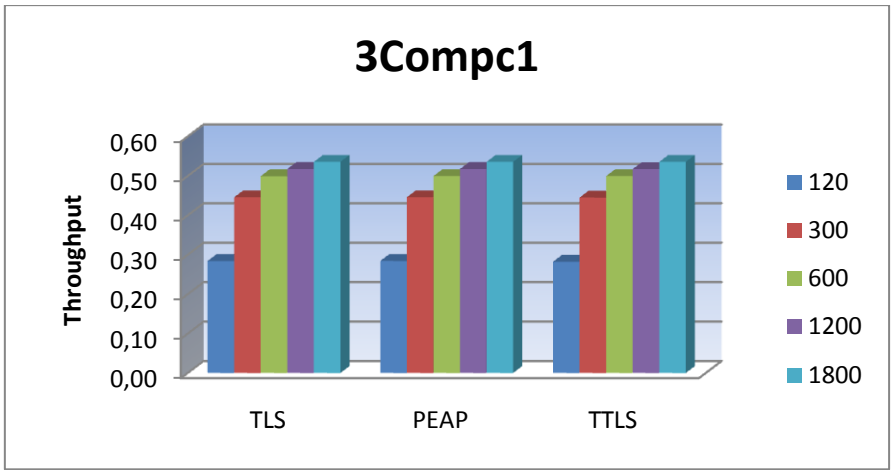


Fig. 4.12 Estudio del escenario 3Com en pc1

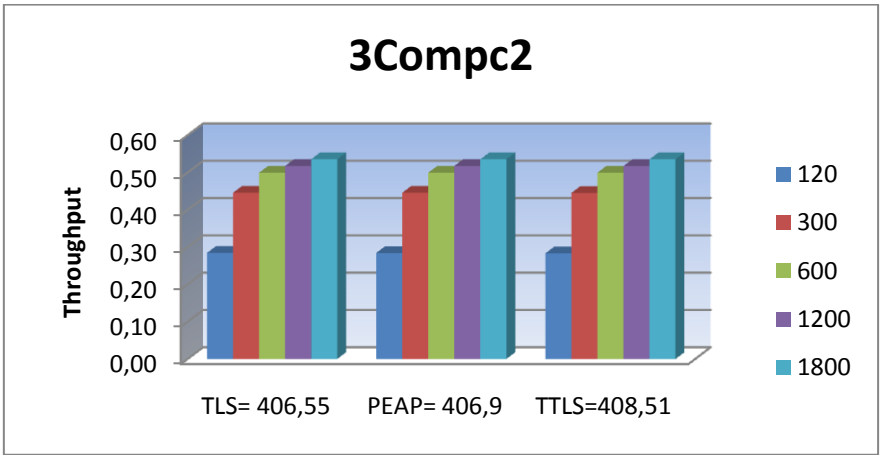


Fig.4.13 Estudio del escenario 3Com en pc2

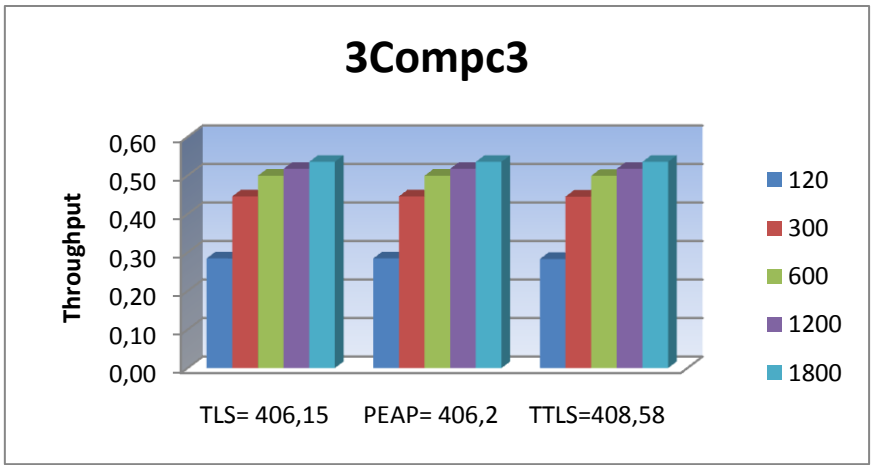


Fig 4.14 Estudio del escenario 3Com en pc3

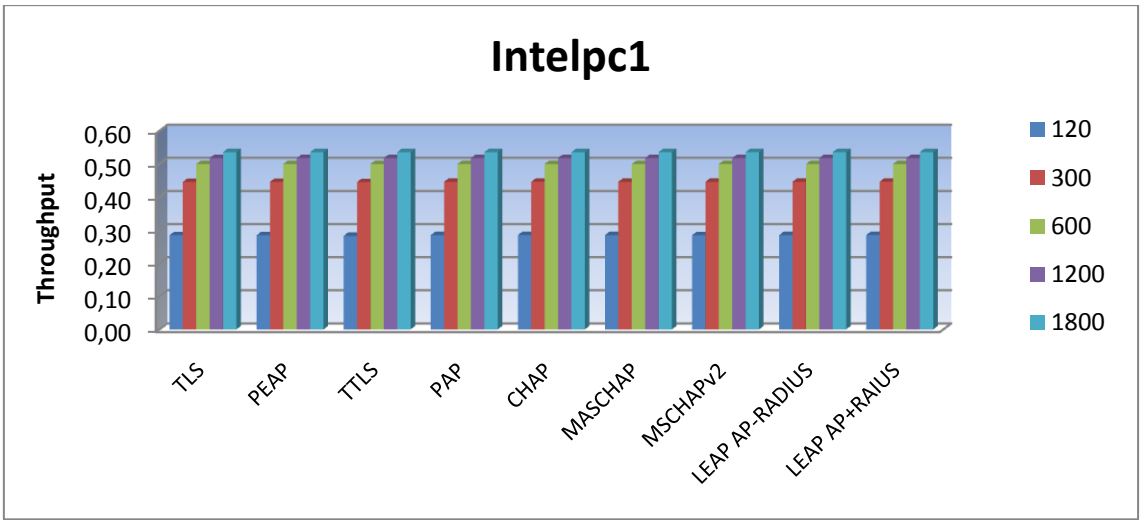


Fig. 4.15 Estudio del escenario Intel en pc1

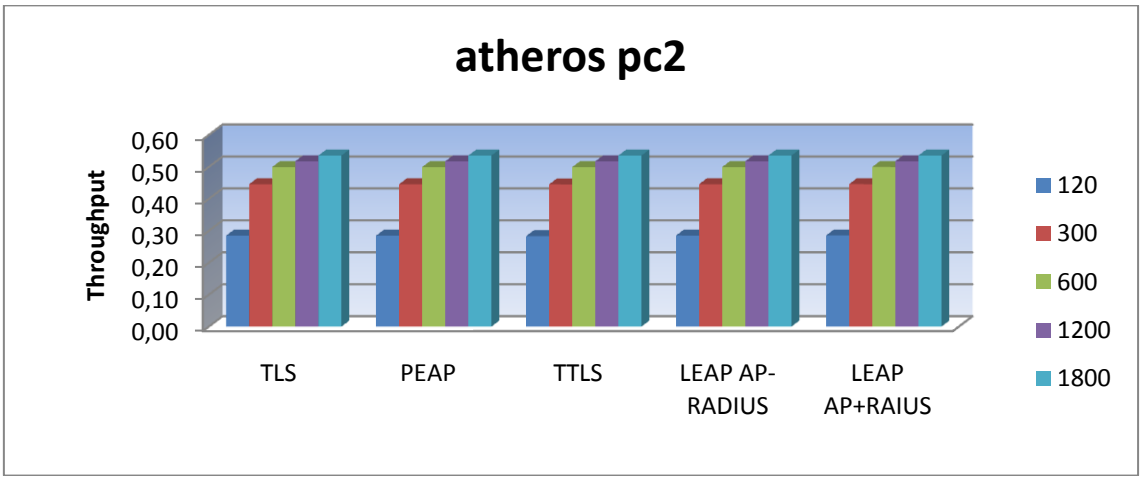


Fig. 4.16 Estudio del escenario Atheros en pc2.

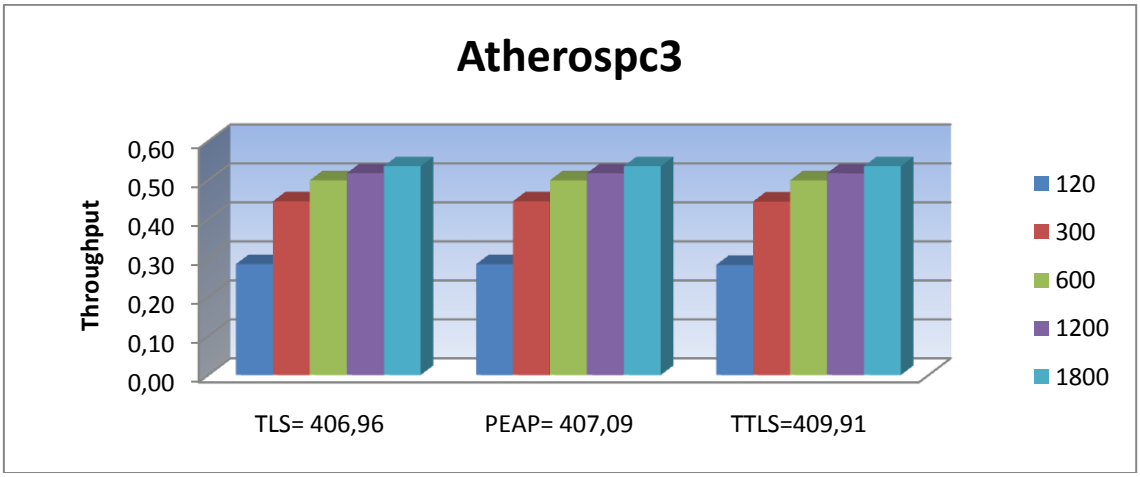


Fig. 4.17 Estudio del escenario Atheros en pc3.

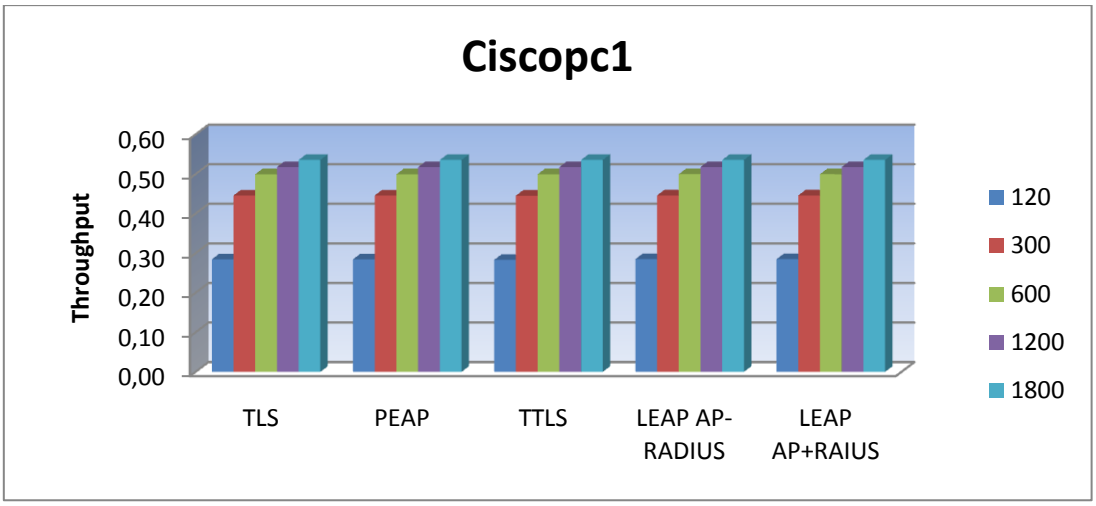


Fig. 4.18 Estudio del escenario Cisco en pc1.

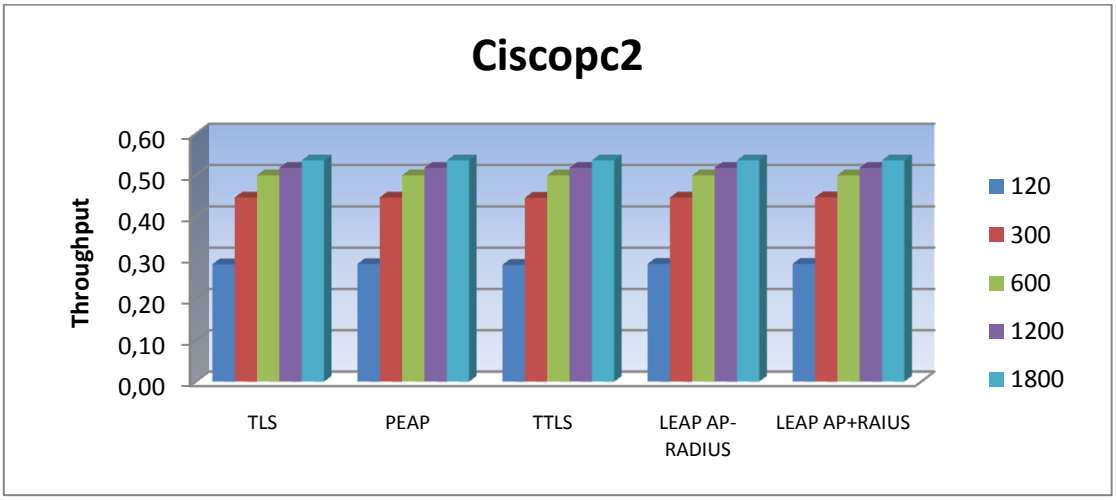


Fig. 4.19 Estudio del escenario Cisco en pc2.

Mediante la tabla 4.14 se resume qué protocolos de autenticación responden mejor frente al ataque para cada uno de los escenarios. Se indican los porcentajes de diferencia entre los protocolos de autenticación con mayor y menor throughput para cada escenario.

Se comprueba que el número de ataques se va reduciendo a medida que el periodo de ataque va aumentando y como consecuencia, el porcentaje de diferencia del throughput disminuye. Por esta razón, se coge como referencia el periodo de ataque más pequeño para poder observar las diferencias más grandes.

Tabla 4.14 Resumen porcentaje de diferencia para periodos de ataque 120,668s.

Escenario	Protocolo de autenticación más óptimo	% mayor de diferencia
3Com en pc1	PEAP	0,73%
3Com en pc2	TLS	0,45%
3Com en pc3	TLS	0,52%
Intel en pc1	LEAP(ap+servidor radius)	0,91%
Atheros en pc2	LEAP(ap+servidor radius)	0,77%
Atheros en pc3	TLS	0,72%
Cisco en pc1	LEAP(ap radius)	0,65%
Cisco en pc2	LEAP(ap+servidor radius)	0,78%

4.2.1.6. Influencia del tipo de escenario en los protocolos de autenticación.

4.2.1.6.1. Plan de pruebas

En esta parte del estudio se analiza cómo influyen los diferentes escenarios en los protocolos de autenticación, de esta forma, se conseguirá averiguar qué escenario tiene el mejor rendimiento para cada uno de los protocolos de autenticación.

Se aplican los diferentes escenarios durante diferentes periodos de ataque con valores de 120.668, 301.67, 603.34, 1206.68, 1800 segundos a cada protocolo de autenticación.

La tabla 4.15 muestra los valores fijados para la realización de las pruebas.

Tabla 4.15 valores fijados para el estudio de los protocolos de autenticación.

Velocidad	54 Mbps
Tiempo de simulación	1800 s
Tamaño del paquete de datos	1500 bytes
Número de STAs	10

4.2.1.6.2. Resultados obtenidos

Con las figuras 4.20, 4.21, 4.22, 4.23 y 4.24, se comprueba que el escenario que presenta el menor retardo de autenticación (véase tabla 4.12), se obtiene el throughput más elevado en cada protocolo de autenticación analizado. La combinación ideal para conseguir el mayor throughput debe emplear el

protocolo de autenticación con el proceso de autenticación más simple/rápido junto con el escenario cuyo rendimiento sea más elevado.

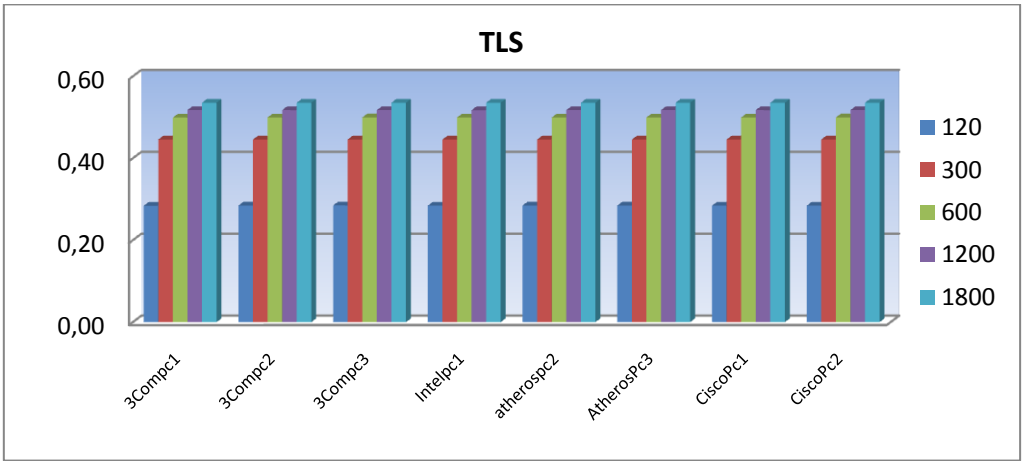


Fig. 4.20 Estudio del protocolo EAP-TLS,

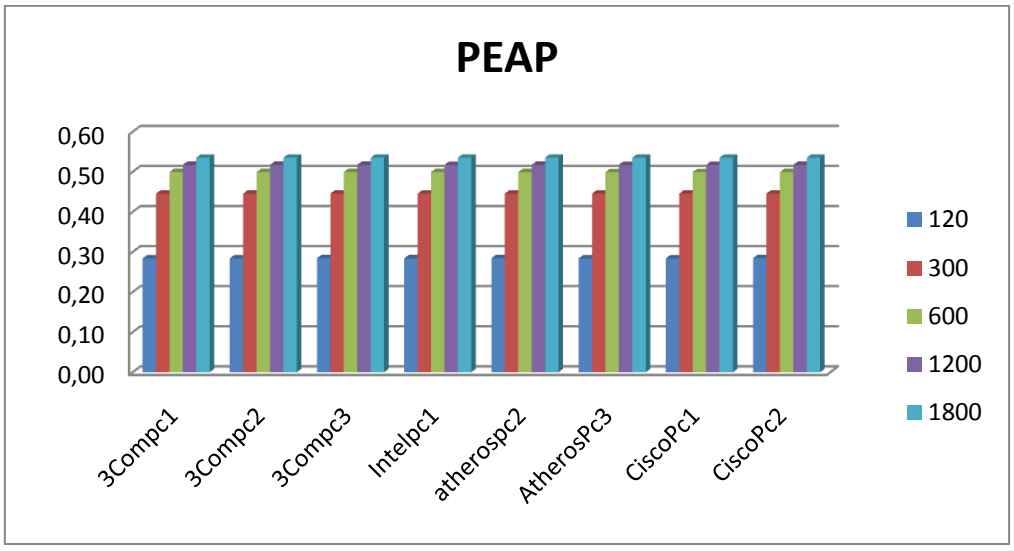


Fig. 4.21 Estudio del protocolo EAP-PEAP

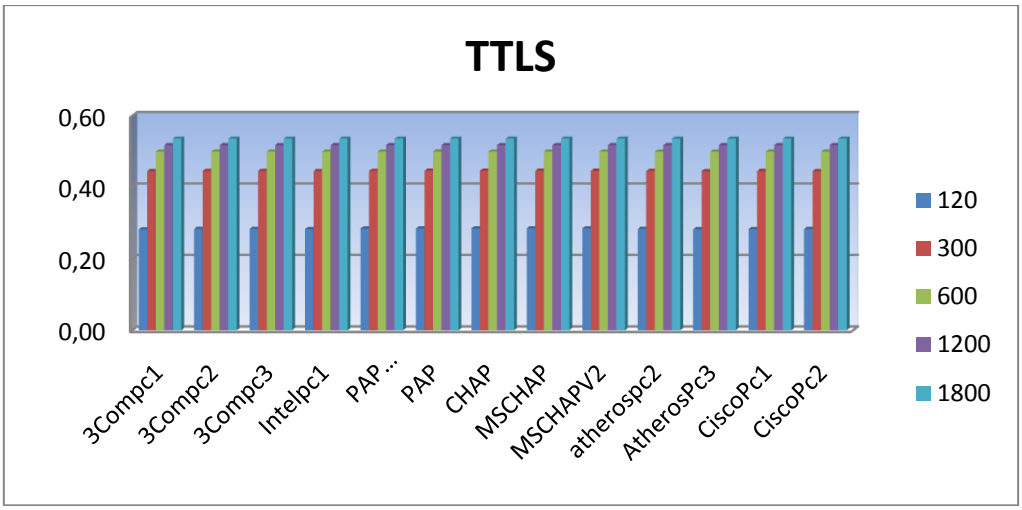


Fig. 4.22 Estudio del protocolo EAP-TTLS.

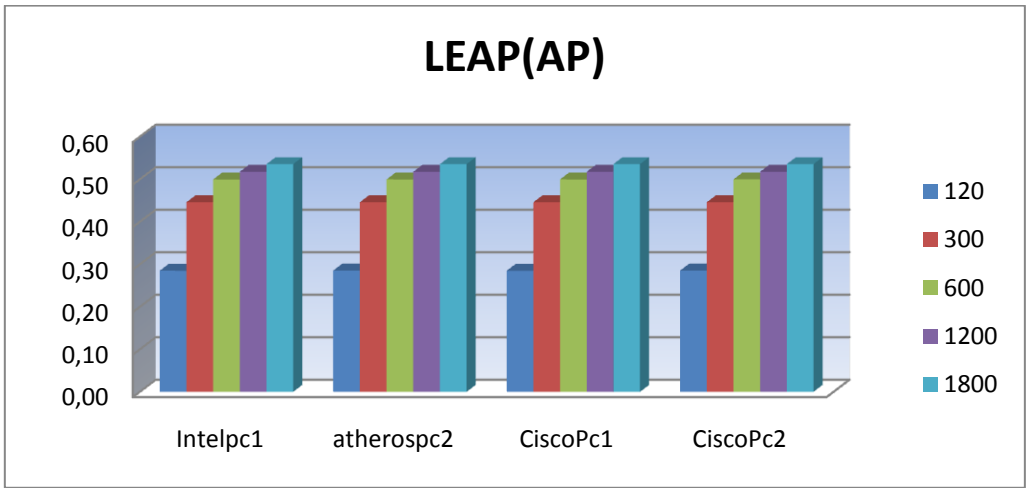


Fig. 4.23 Estudio del protocolo EAP-LEAP

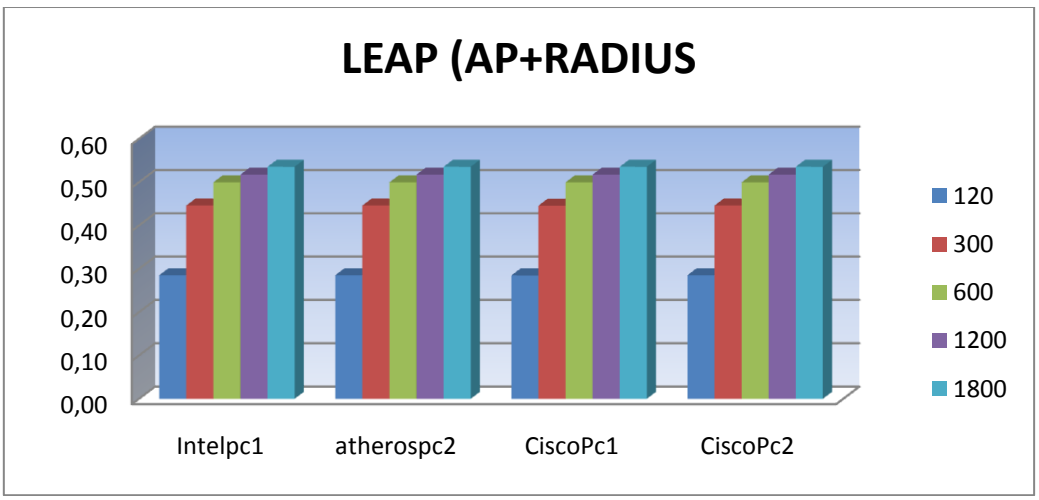


Fig. 4.24 Estudio del protocolo EAP-LEAP (AP+ servidor RADIUS)

La tabla 4.16 muestra el incremento porcentual respecto al escenario que proporciona peor comportamiento para cada protocolo de autenticación.

Tabla 4.16 Resumen de las tarjetas más optimas para cada protocolo.

Protocolo de autenticación	Escenario	% mayor de diferencia
TLS	3Com en pc3	0,20%
PEAP	3Com en pc3	0,20%
TTLS	3Com en pc2	0,19%
TTLS (con software intel)	Intel en pc1 con CHAP	0,97%
LEAP (AP sin servidor Radius)	Atheros en pc2	0,19%
LEAP (AP con servidor Radius)	Atheros en pc2	0,19%

En nuestro estudio, el valor máximo de throughput obtenido lo conseguimos con el protocolo LEAP (AP+ servidor RADIUS) y el escenario Atheros en pc2 con el periodo de ataque más grande. Con esta tarjeta y este protocolo, el tiempo de autenticación es más pequeño y por lo tanto el tiempo perdido durante el ataque también. Esto se traduce en un aumento del throughput.

4.2.2. Modelo analítico

En este apartado compararemos los resultados de las pruebas anteriores con los resultados obtenidos a partir de un modelo analítico derivado de dos modelos anteriores: el modelo de Bianchi [1] y el modelo de analítico extraído de la Tesis doctoral de Elena López [2].

4.2.2.1. Modelo de Bianchi

El modelo publicado por G.Bianchi [1], presenta una evaluación precisa sobre el throughput de saturación en redes IEEE 802.11 DCF.

Bianchi define el throughput de saturación normalizado *S* mediante la ecuación (4.1).

$$S = \frac{P_{tr} \cdot P_s \cdot E_p}{r_{tx} \cdot E_S} \quad \textbf{(4.1)}$$

Donde E_S es la duración media de una ranura, $\mathrm{r_{tx}}$ es la tasa de transmisión de datos, $\mathrm{E_p}$ es la longitud de datos de la trama, $\mathrm{P_{tr}}$ representa la probabilidad de que almenos una estación transmita y $\mathrm{P_s}$ es la probabilidad que una transmisión concluya con éxito.

4.2.2.2. Modelo analítico Propuesto

El modelo de G.Bianchi sigue siendo aplicable aunque se produzca la desautenticación, sólo cambia el tiempo medio de slot E_S por E_S' :

$$S = \frac{P_{tr} \cdot P_S \cdot E_p}{r_{tx} \cdot E_s'} \quad \textbf{(4.2)}$$

$$E_s' = \frac{1/\lambda}{\frac{1}{\lambda} - T_{ocupado_{ataque}}} \cdot E_S \quad \textbf{(4.3)}$$

Donde $1/\lambda$ es el periodo del ataque, E_S es el empleado en el modelo de Bianchi y el $T_{ocupado_{ataque}}$, que es el tiempo añadido a la ocupación del canal por el ataque y vendrá determinado dependiendo si la desautenticación se produce mientras el canal está libre u ocupado.

$$T_{ocupado_{ataque}} = P_{ocupado} \cdot T_{ocupado} + P_{libre1} \cdot T_{libre_1} + P_{libre_2} \cdot T_{libre_2} \quad \textbf{(4.4)}$$

Si el canal está ocupado, éste lo estará con una probabilidad $P_{ocupado}$.

$$P_{ocupado} = \frac{P_{tr} \cdot P_s \cdot (T_s - DIFS) + P_{tr}\,(1 - P_S) \cdot (T_c - DIFS)}{E_s} \quad \textbf{(4.5)}$$

La probabilidad de que el canal esté ocupado puede venir dada o por una transmisión o por una colisión.

Los tiempos T_c y T_s corresponden respectivamente al periodo de tiempo en el cual se encuentra ocupado el canal con un transmisión con éxito y con una colisión. El tiempo medio de slot E_s, la probabilidad de que al menos una estación esté transmitiendo P_{tr} y la probabilidad de una transmisión con éxito P_s son los mismos valores originales del modelo de G. Bianchi. Para el cálculo de P_{tr} y P_s se programaron dos funciones que se pueden consultar en el anexo E.

Consideramos que el AP desautentica una vez que el medio ha quedado desocupado. El tiempo que el AP emplea para la desautenticación es el siguiente:

$$Tocupado = T_{ataque} \quad \textbf{(4.6)}$$

Donde el tiempo de ataque consiste en la suma de los 60s y el tiempo de desautenticación.

$$T_{ataque} = 60s + T_{autenticación} \quad \textbf{(4.7)}$$

Si el canal está libre puede darse por dos casos, el primero durante un tiempo DIFS y el segundo durante un tiempo de backoff.

Si el canal está libre durante un tiempo DIFS, este evento ocurrirá con una probabilidad y una duración:

$$P_{libre_1} = \frac{P_{tr} \cdot DIFS}{E_s} \quad \textbf{(4.8)}$$

$$T_{libre_1} = 0.5\ DIFS + Tataque \quad \textbf{(4.9)}$$

Donde $0.5\ DIFS$ corresponde a la media del tiempo DIFS que ha pasado.

Cuando se encuentra el canal libre por un slot de backoff significa que las STAs están disminuyendo su contador de backoff. Esto ocurre con una probabilidad y duración:

$$P_{libre_2} = \frac{(1-P_{tr}) \cdot \sigma}{E_s} \quad \textbf{(4.10)}$$

$$T_{libre_2} = \frac{\sigma}{2} + tiempo\ de\ ataque + DIFS \quad \textbf{(4.11)}$$

Donde $\frac{\sigma}{2}$ es la media del tiempo de slot (σ) que ha pasado y el tiempo DIFS que esperan las estaciones después de una autenticación antes de volver a transmitir.

La tabla 4.17 muestra los resultados del throughput obtenidos con el simulador 802.11 (Ssimulador), contrastándolos con el modelo anteriormente estudiado (Sanalítico). Se muestra los resultados para el periodo de ataque 120,668s, el resto de periodos los podemos ver en el anexo F.

Tabla 4.17 Comparativa de los resultados obtenidos mediante la simulación con los observados para el modelo analítico.

Número de STAs	Tamaño de la trama	Sanalítico	Ssimulado	% diferencia
2	200	0,074	0,092	19,030
4	200	0,077	0,092	16,066
10	200	0,077	0,088	13,085
15	200	0,075	0,086	12,681
20	200	0,074	0,084	12,180
50	200	0,068	0,077	11,805

100	200	0,062	0,070	11,805
2	600	0,166	0,204	18,564
4	600	0,172	0,199	13,882
10	600	0,165	0,189	12,732
15	600	0,160	0,181	11,720
20	600	0,156	0,176	11,355
50	600	0,141	0,159	11,263
100	600	0,127	0,143	11,359
2	1000	0,227	0,269	15,564
4	1000	0,230	0,259	11,476
10	1000	0,217	0,243	10,806
15	1000	0,210	0,232	9,857
20	1000	0,204	0,225	9,550
50	1000	0,183	0,202	9,581
100	1000	0,163	0,181	9,718
2	1500	0,274	0,323	15,203
4	1500	0,271	0,308	11,869
10	1500	0,253	0,284	10,816
15	1500	0,243	0,273	10,985
20	1500	0,236	0,264	10,650
50	1500	0,210	0,235	10,878
100	1500	0,187	0,210	11,102

Como se puede observar los porcentajes de error se encuentran alrededor de un 10%. En el anexo F, se presentan los resultados para periodos de ataque superiores; el porcentaje de diferencia entre los resultados simulados y analíticos disminuye.

4.2.3. Las STAs son atacadas

En esta sección se tiene en cuenta el escenario 2 donde se elijen las estaciones que son atacadas y se establece que para cada periodo de ataque, las STAs atacadas no puedan transmitir por un tiempo de ataque.

Primeramente estudiaremos diferentes escenarios con una estación atacada y después analizaremos el incremento del número de STAs atacadas.

En esta parte del estudio se distingue el throughput medio de las STAs atacadas y el throughput medio de las STAs no atacadas.

4.2.3.1. 1 STA atacada

4.2.3.1.1. Influencia del número de STAs y el periodo del ataque para el escenario Intel en pc1

4.2.3.1.1.1. Plan de pruebas

En este apartado se pretende analizar cómo influye la variación del periodo de ataque y el número de STAs enviando paquetes de 200 y 1500 bytes.

Para llevar a cabo estas pruebas se ha cogido como referencia el escenario 3Com en Pc1 con el protocolo de autenticación EAP-TLS, en el anexo G se muestra un estudio de este escenario con los protocolos de autenticación restantes.
A cada paquete de datos de 200 y 1500 bytes se les aplica variaciones de 2, 4, 10, 15, 20, 50 y 100 STAs con periodos de ataque de 120,668 s,150s, 181,002s, 210s, 301, 67 s, 603,34 s, 1206,68 s y 1800s.

4.2.3.1.1.2. Resultados obtenidos

Se comprueba que el throughput de las STAs atacadas se incrementa con el aumento del periodo de ataque, en cambio, el throughput de las STAs no atacadas disminuye conforme aumentamos el periodo de ataque. Además, a medida que se incrementa el número de estaciones y el periodo de ataque, el throughput de las STAs atacadas y el throughput de las STAs no atacadas tienden a igualarse.

Como se puede observar en la figura 4.25, a partir de los 150s de periodo de ataque los throughputs se igualan, de esta forma, en las pruebas restantes del trabajo se analizan periodos de ataque de 90, 120 y 150 segundos para poder apreciar con más claridad los resultados. Estos efectos se producen porque a medida que la estación atacada tiene más oportunidades de transmitir, menos probabilidades tendrán las estaciones no atacadas, y a medida que aumenta el número de estaciones, menos probabilidades tienen las estaciones de transmitir.

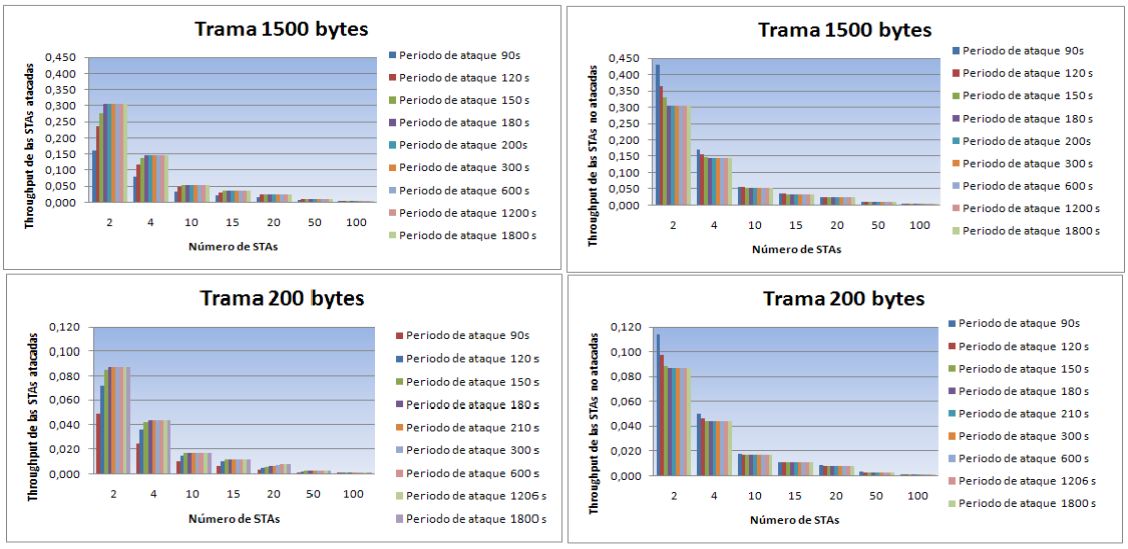


Fig. 4.25 Influencia del número de STAs y periodo de ataque para 3Com en pc1

Junto con el estudio del anexo G, la tabla 4.18 muestra los porcentajes de diferencia más significativos entre throughput de las STAs atacadas y el throughput de las STAs no atacadas para cada uno de los protocolos de autenticación. Se coge como referencia el periodo de ataque y el número de STAs más pequeño (periodo de ataque 90s y 2 STAs), ya que con esta combinación se contemplan las diferencias de throughput más reveladoras.

Tabla 4.18 Diferencia entre el throughput de las STAs atacadas y no atacadas.

	EAP -TLS	EAP-PEAP	EAP-TTLS (incluído software Intel)	EAP- LEAP (AP radius)	EAP- LEAP (AP+ servidor radius)
Trama 200 bytes	56,14%	56,14%	56’96%	55,93%	56,14%
Trama 1500 bytes	62,47%	62,23%	63,48%	62,17%	62,20%

Como se puede ver, los resultados más significativos se obtienen con el protocolo EAP-TTLS y con el escenario de la tarjeta Intel y el Pc1 con el método CHAP. Se consigue un 63,48% enviando la trama más grande de datos con el periodo de ataque más pequeño y el menor número de estaciones. En todos los casos mostrados, el throughput de las STAs no atacadas es mayor que el de las atacadas.

4.2.3.1.2. Influencia del número de STAs atacadas.

En esta parte del estudio analizaremos la influencia del incremento del número de STAs atacadas para el protocolo EAP-TLS. Se efectuarán las pruebas del protocolo EAP-TLS con periodos de ataque de 90s, 120s y 150s, enviando tramas de 200 y 1500 bytes.

4.2.3.1.2.1. Estudio del protocolo EAP-TLS con 10 STAs atacadas.

4.2.3.1.2.1.1. Plan de pruebas

En este apartado se analizará la influencia del número de STAs atacadas en el protocolo EAP-TLS para un escenario de 10 STAs. Se aumentara de una en una el número de STAs atacadas desde 0 hasta 10. Además, se estudiará cómo afecta el periodo de ataque y se envían tramas de 1500 bytes de datos.

4.2.3.1.2.1.2. Resultados obtenidos

Los resultados que muestran las figuras 4.26 - 4.33 demuestran que, el comportamiento de los escenarios es el mismo obteniendo así resultados muy similares.

Partiendo del escenario donde todas las STAs están siendo atacadas y el periodo de ataque es de 90s (escenario más crítico para las STAs atacadas), se puede observar que el throughput de las STAs atacadas aumenta en un 40 % cuando se incrementa el periodo de ataque hasta los 150s. Sin embargo, si tomamos como escenario 9 STAs atacadas y una sin atacar con un periodo de ataque de 90s (el mejor escenario para una STA no atacada), se observa como el throughput de las STAs no atacadas decrece un 40 % al incrementar el periodo de ataque hasta los 150s.

Este evento indica que, a medida que se aumenta el periodo de ataque los throughputs tienden a igualarse. Esto se produce al decrementar el número de ataques sobre las STAs atacadas, obteniendo como resultado, que tanto las STAs atacadas como las no atacadas tengan la misma probabilidad de transmitir.

Resaltar como en la figura del throughput de las STAs no atacadas se obtiene el throughput máximo de 0,09 con el escenario 3Com en pc3 cuando 9 de las 10 estaciones están siendo atacadas, consiguiendo una diferencia del 64 % entre las STAs atacadas y las que no lo están.

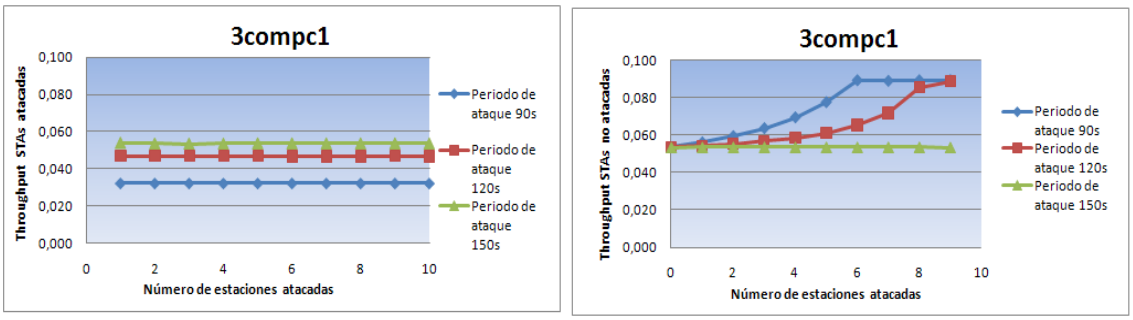


Fig. 4.26 Estudio del throughput para el escenario 3Com en pc1

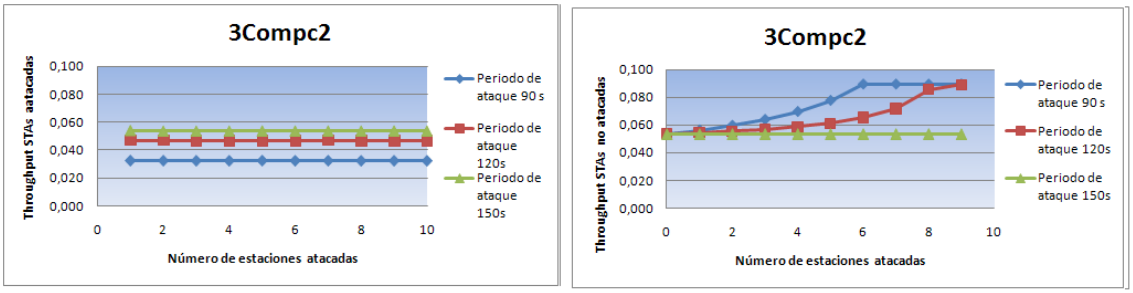


Fig. 4.27 Estudio del throughput para el escenario 3Com en pc2.

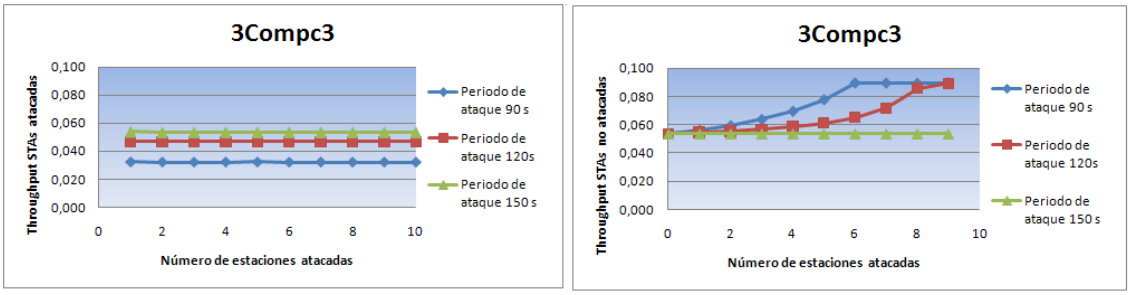


Fig. 4.28 Estudio del throughput para el escenario 3Com en pc3.

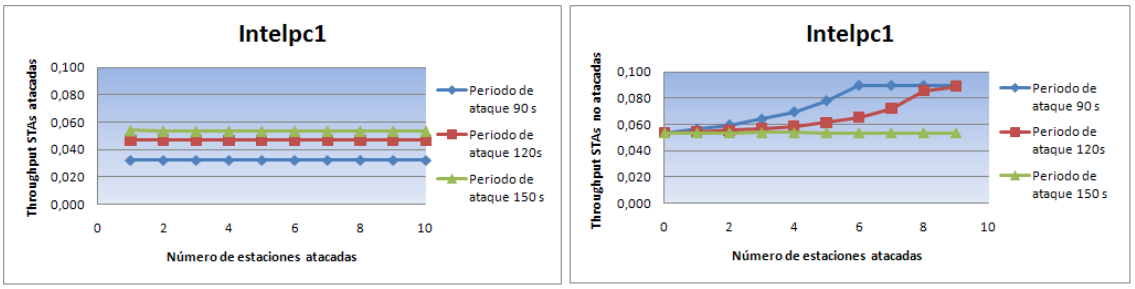


Fig. 4.29 Estudio del throughput para el escenario Intel en pc1

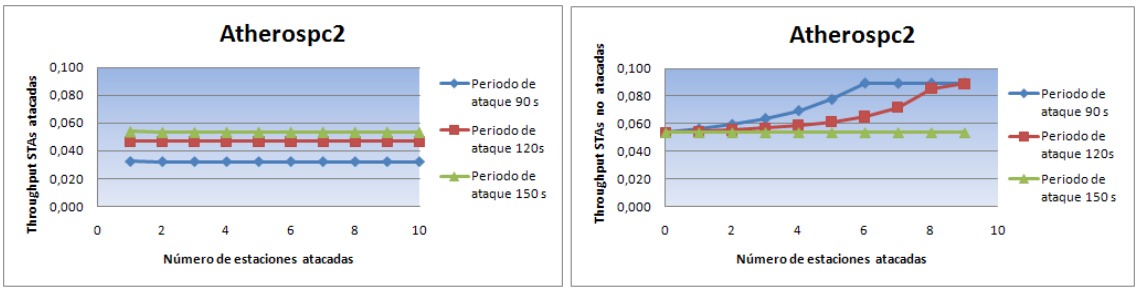


Fig. 4.30 Estudio del throughput para el escenario Atheros en pc2.

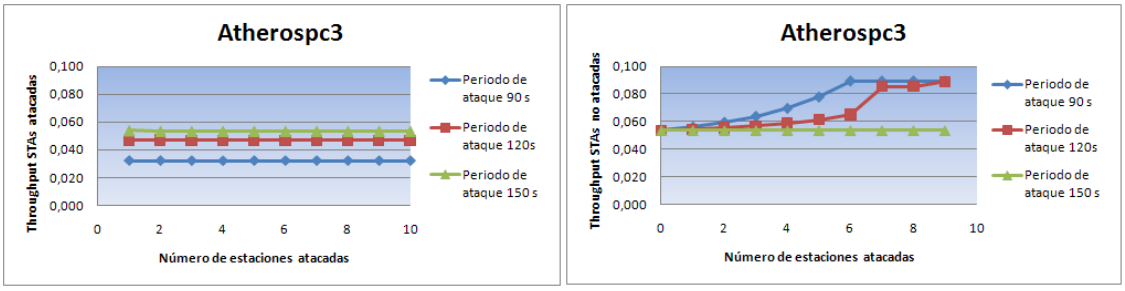


Fig. 4.31 Estudio del throughput para el escenario Atheros en pc3.

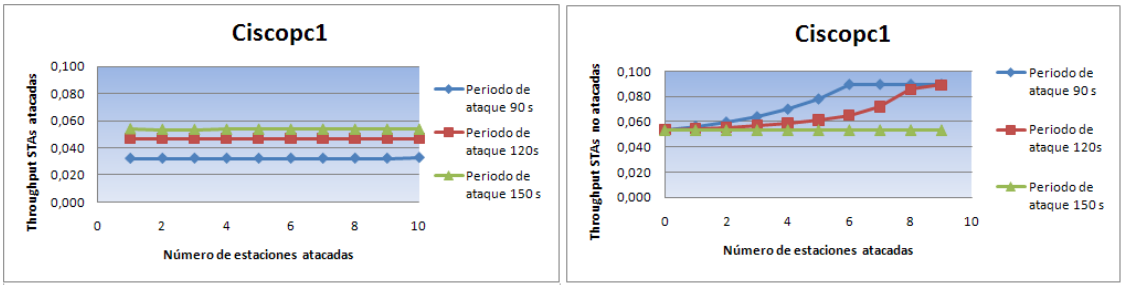


Fig. 4.32 Estudio del throughput para el escenario Cisco en pc1.

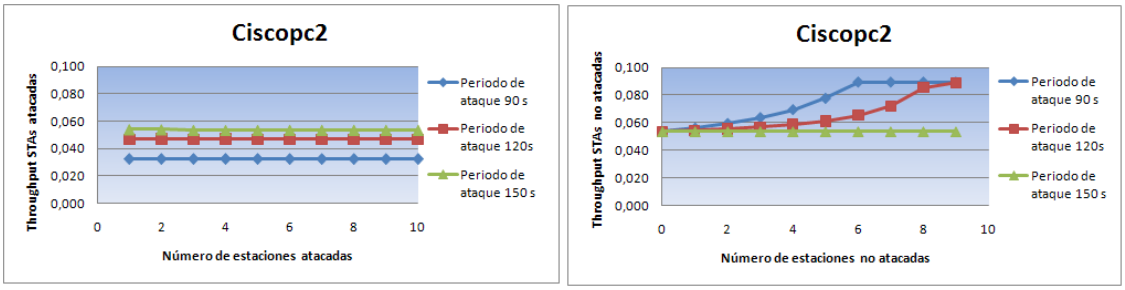


Fig. 4.33 Estudio del throughput para el escenario Cisco en pc2.

4.2.3.1.2.2. Influencia del número de estaciones atacadas

4.2.3.1.2.2.1. Plan de pruebas

En este apartado se investiga cómo influye en el throughput el incremento del número de estaciones atacadas. Se empleará el escenario Intel en Pc1 con el protocolo EAP-TLS y se analizarán tres escenarios, al primero se irá incrementando el número de estaciones atacadas de una en una desde 0 hasta 10, en el segundo se hace lo mismo que en el anterior pero de 0 hasta 20, y en el tercer escenario, se incrementarán de dos en dos desde 0 hasta 50 STAs. Para cada uno de los escenarios se aplicarán periodos de ataque de 90s, 120s y 150s, enviando paquetes de 200 y 1500 bytes.

4.2.3.1.2.2.2. Resultados obtenidos

Como se puede comprobar con las figuras 4.34 y 4.35, si se aumenta el número de estaciones, el throughput de las STA atacadas y el throughput de las STA no atacadas decrece.

A medida que se incrementa el número de estaciones atacadas, el throughput de las STAs atacadas permanece constante y aumentando el periodo de ataque, éste se incrementa. Partiendo del escenario donde todas las STAs están siendo atacadas con un periodo de ataque de 90s, observamos que al aumentar el periodo de ataque hasta los 150s, el throughput de las STAs atacadas crece. La tabla 4.19 muestra el porcentaje de este incremento para

los escenarios evaluados, concluyendo que, el throughput de las STAs atacadas se incrementa alrededor de un 40% independientemente del tamaño de la trama y del número de STAs.

Tabla 4.19 Porcentajes del aumento del throughput de las STAs atacadas.

Número de STAs	Trama 200 bytes	Trama 1500 bytes
10 STAs	42,32%	41,03%
20 STAs	42,47%	38,97%
50 STAs	41,56%	40,23%

Por otra parte, el throughput de las STAs no atacadas disminuye a medida que el periodo de ataque va aumentando hasta que se iguala con el throughput de las STAs atacadas. Esta tendencia a igualarse se produce con más rapidez con la trama de 1500 bytes.

Mientras aumenta el número de estaciones atacadas, el throughput de las STAs no atacadas va aumentando de forma rápida hasta llegar a un punto donde para de crecer. Este evento se produce cuando la mayoría de estaciones están siendo atacadas y las pocas que no lo están, no paran de transmitir. El protocolo 802.11 evita que una estación monopolice el canal de transmisión añadiendo tiempos de backoff entre transmisiones consecutivas, por lo tanto, la interrupción del aumento del throughput de las STAs no atacadas que se muestra es debido a estos tiempos de backoff, además de los tiempos DIFS, SIFS y los bytes de cabecera.

Destacar que, para la trama de 1500 bytes el crecimiento del throughput de las STAs no atacadas es más lento que en el caso de las tramas de 200 bytes y por lo tanto, menos tiempos de backoff se añadirán. Esto ocurre porque al ser mayor el tamaño de datos a enviar en cada transmisión, no hacen falta tantas transmisiones para enviar la información.

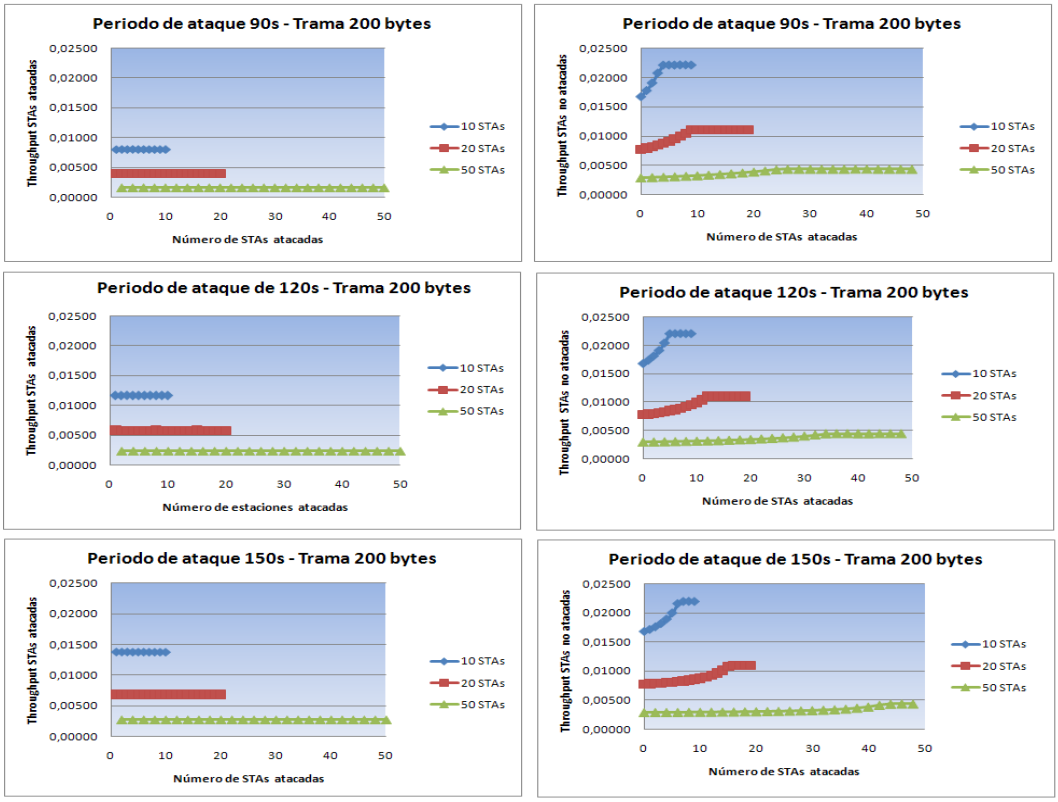


Fig. 4.35 Influencia del aumento de número de STAs atacadas con 200 bytes.

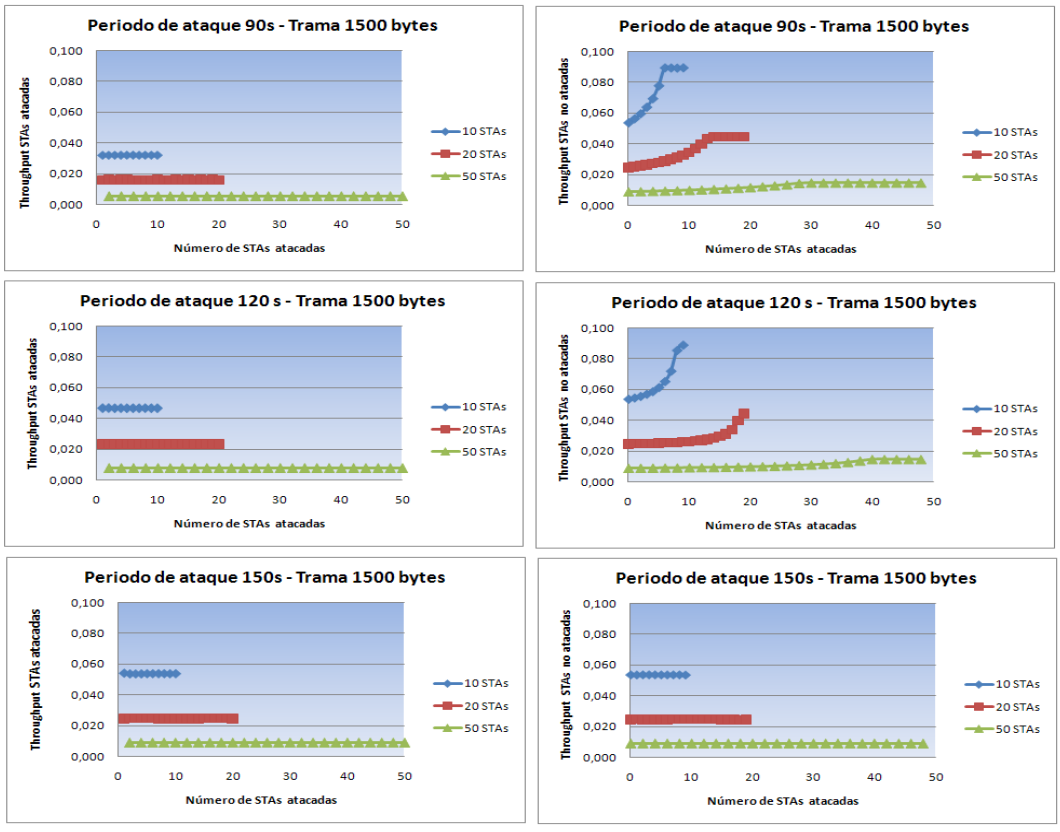


Fig. 4.36 Influencia del aumento de número de STAs atacadas con 1500 bytes.

4.3. Estudio de la justicia

En este apartado se evaluará la justicia de acceso al canal mediante el índice de justicia Jain Index.

Considerando *N* el número de estaciones en el sistema y Y_i sea la fracción de las transmisiones realizadas por la STA *i* durante la ventana *W*, el índice de justicia es el siguiente:

$$\mathbf{F_j\ (W) = \frac{(\sum_{i=1}^{N} Y_i)^2}{N\ \sum_{i=1}^{N} Y_i^2}}$$

Si el Jain Index toma valor de 1, se logra una justicia perfecta, sin embargo, si el valor es de $1/N$, siendo *N* el número de estaciones, se dice que la justicia es imperfecta.

Normalizamos el tamaño de la ventana respecto al número de estaciones y calculamos el Jain Index para tamaños de ventana que son múltiplos de de *N*. Utilizaremos un tamaño de ventana normalizada *m* de tal manera que *W = m X N* y m=0, 1,2,….

Se realizan dos estudios, en el primero, se analiza la justicia en el protocolo EAP-TLS y el segundo, se evalúa cómo influye en la justicia el número de STAs atacadas. En los dos casos, se tomará como referencia el escenario Intel en Pc1.

4.3.1. Estudio de la justicia en el protocolo EAP-TLS

4.3.1.1. Plan de pruebas

En esta apartado analizaremos la justicia del acceso al canal con el protocolo EAP-TLS. Se incrementa de uno en uno el número de estaciones atacadas desde 0 hasta 10 y se aplican periodos de ataque de 90, 120 y 150 segundos, enviando tramas de 1500 bytes.

En los resultados obtenidos se muestran las figuras para el escenario Intel en Pc1 elegido como escenario de referencia, el resultado de los escenarios restantes se pueden consultar en el anexo H.

4.3.1.2. Resultados obtenidos

Hemos podido observar con los resultados obtenidos que conseguimos una mayor justicia a largo plazo.

Como se observa en las figuras 4.37, 4.38 y 4.39, al aumentar el periodo de ataque el índice de justicia aumenta.

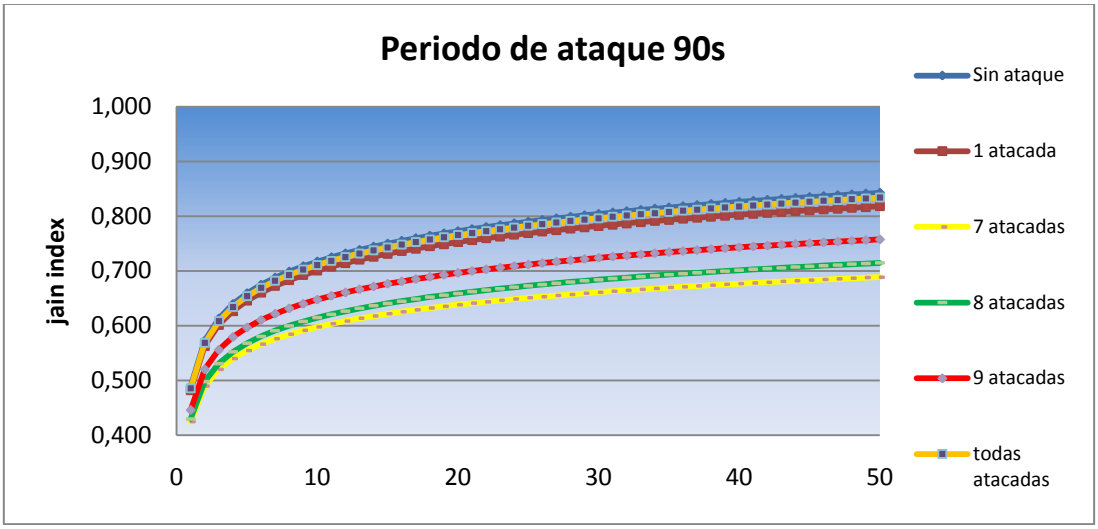


Fig. 4.37 10 STAs atacadas con periodo de ataque 90s y trama 1500 bytes.

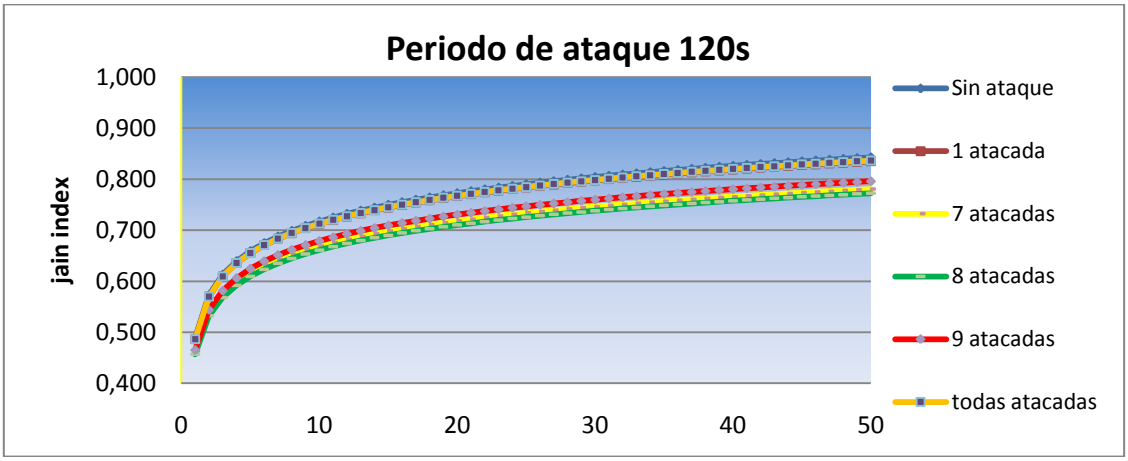


Fig. 4.38 10 STAs atacadas con periodo de ataque 120s y trama 1500 bytes.

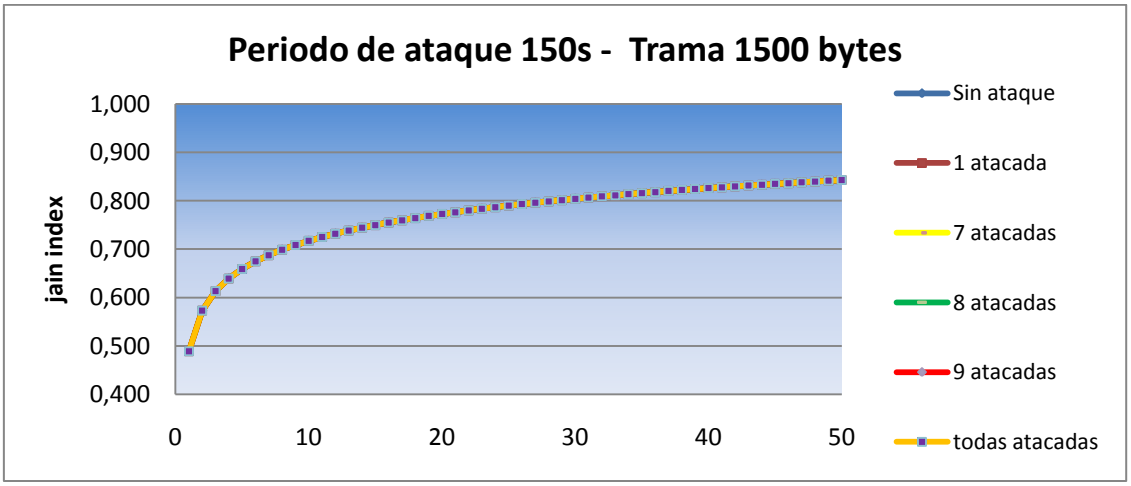


Fig. 4.39 10 STAs atacadas con periodo de ataque 150s y trama 1500 bytes.

Conforme aumenta el número de estaciones atacadas, la justicia decrece, es decir, el Jain Index disminuye. Este último, no permanece disminuyendo constantemente si no que llega un punto en el cual para de disminuir y empieza a subir de nuevo. Esto es debido a que las STAs no atacadas son minoritarias y el protocolo de la red 802.11 aplica tiempos de backoff entre transmisiones consecutivas para evitar que algunas monopolicen el medio.

De esta forma, como se observa en la figura 4.37, en el punto donde se ataca a 7 de las 10 estaciones, el Jain Index empieza de nuevo a incrementarse con la tendencia a equipararse con el índice más justo (punto de inflexión).

En la tabla 4.20 se resume los porcentajes de diferencia entre el Jain Index más grande y más pequeño para cada periodo de ataque, indicando además el punto de inflexión donde se produce el incremento de justicia causados por los mecanismos de acceso a la red.

Tabla 4.20 Porcentaje de diferencia del Jain Index.

Periodo de ataque	% de dieferencia	Punto de inflexión
90s	12,26%	7 estaciones atacadas
120s	6,30%	8 estaciones atacadas
150s	Todos con el mismo índice de justicia	Índice de justicia constante

Comparando con los resultados del anexo H, el escenario que más diferencia de porcentaje presenta entre sus índices de justicia es 3Com en pc1 con un periodo de ataque de 90s, de esta forma, se obtiene un13, 29% de diferencia.

Los valores más representativos los obtenemos con el periodo de ataque de 90s, ya que es el periodo con el que se producen más ataques y es donde la justicia se ve más comprometida.

4.3.2. Influencia del número de STAs atacadas en la justicia

4.3.2.1. Plan de pruebas

En este apartado se evaluará como influye el aumento del número de estaciones atacadas en la justicia del AP con el escenario Intel en Pc1 con el protocolo de autenticación EAP-TLS. Se analizarán tres escenarios diferentes, en el primero se aumenta de uno en uno el número de estaciones atacadas desde 0 hasta 10, en el segundo desde 0 hasta 20 y en el tercero desde 0 hasta 50. Para cada uno de los escenarios se aplicarán periodos de ataque de 90, 120 y 150 segundos, enviando tramas de 200 y 1500 bytes.

4.3.2.2. Resultados obtenidos

Como se pueden ver en la figuras 4.40, 4.41 y 4.42, el incremento del periodo del ataque hace que el Jain Index aumente.

En la tablas 4.21, 4.22 y 4.23, se resume los porcentajes de diferencia entre el Jain Index más grande y más pequeño para cada periodo de tiempo, indicando además el punto de inflexión donde se produce el incremento de justicia causado por los mecanismos de acceso a la red. Este punto de inflexión nos indicará el número de estaciones atacadas a partir del cual empezará a equilibrarse el Jain Index.

Tabla 4.21 Tabla de diferencia de Jain Index para 10 STAs atacadas.

10STAs atacadas						
Tamaño de la trama	Tiempo de ataque 90s		Tiempo de ataque 120s		Tiempo de ataque 150s	
	% diferencia	Punto de inflexión (STAs atacadas)	% diferencia	Punto de inflexión (STAs atacadas)	% diferencia	Punto de inflexión (STAs atacadas)
200 bytes	12,70	7	7,58	7	5,32	7
1500 bytes	13,08	7	6,54	8	0	-

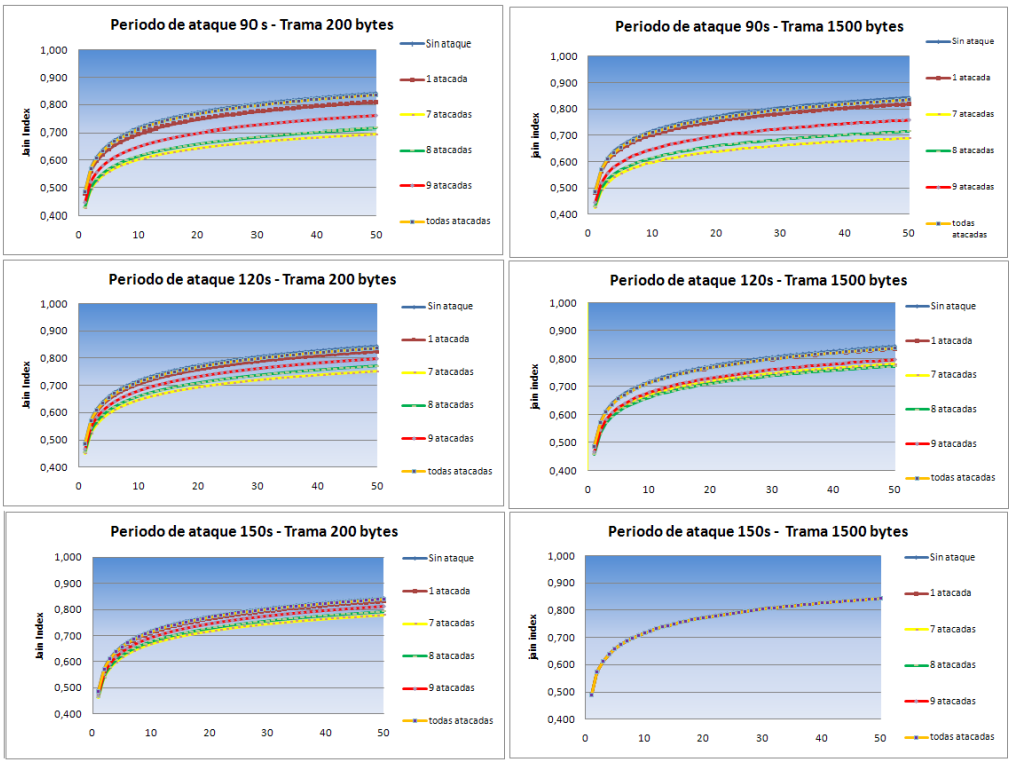


Fig. 4.40 Justicia para 10 STAs atacadas

Tabla 4.22 Tabla de diferencia de Jain Index para 20 STAs atacadas.

20STAs atacadas						
Tamaño de la trama	Tiempo de ataque 90s		Tiempo de ataque 120s		Tiempo de ataque 150s	
	% diferencia	Punto de inflexión (STAs atacadas)	% diferencia	Punto de inflexión (STAs atacadas)	% diferencia	Punto de inflexión (STAs atacadas)
200 bytes	9,84	14	6,91	12	4,52	15
1500 bytes	11,43	14	3,19	18	0	-

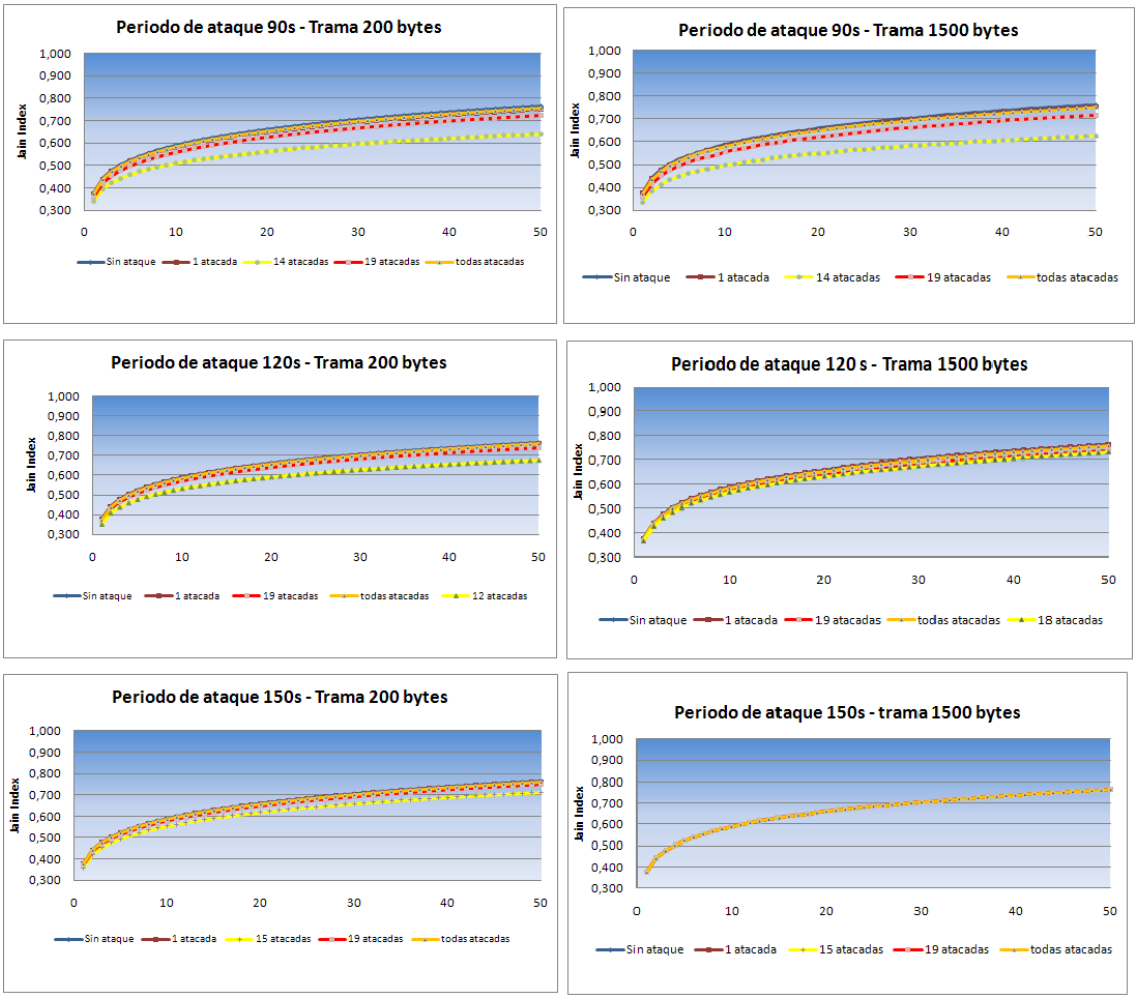


Fig. 4.41 Justicia para 20 estaciones atacadas

Tabla 4.23 Tabla de diferencia de Jain Index para 50 STAs atacadas.

50estaciones atacadas						
Tamaño de la trama	Tiempo de ataque 90s		Tiempo de ataque 120s		Tiempo de ataque 150s	
	% diferencia	Punto de inflexión (STAs atacadas)	% diferencia	Punto de inflexión (STAs atacadas)	% diferencia	Punto de inflexión (STAs atacadas)
200 bytes	10,62	24	6,96	34	3,29	42
1500 bytes	11,35	28	5,86	40	0	-

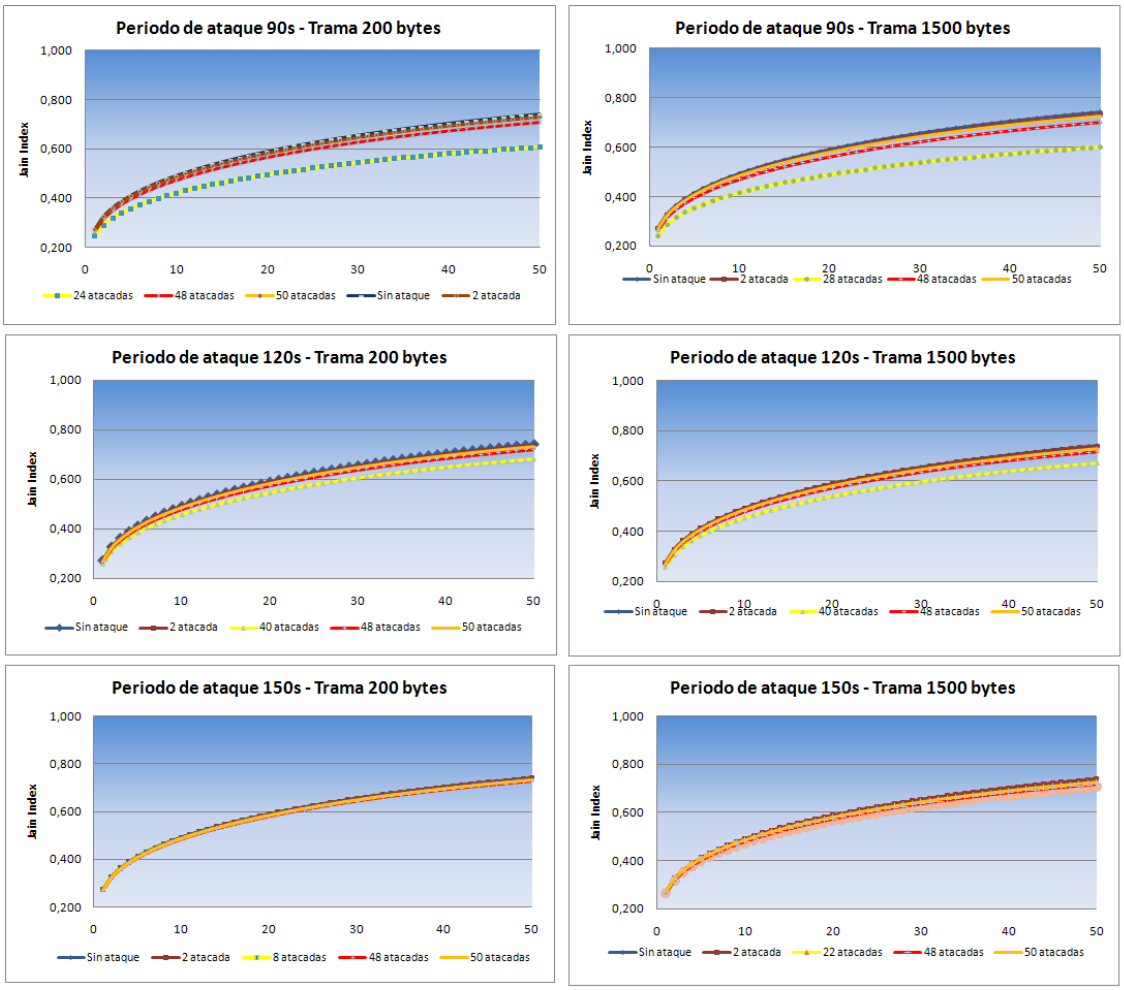


Fig. 4.42 Justicia para 50 STAs atacadas.

Como se puede observar la tarjeta Intel padece más injusticia con el escenario de 10 estaciones y el periodo de ataque de 90s, obteniendo un 13,08% de diferencia entre el Jain Index más justo y el Jain Index más injusto.

Se comprueba que aumentando el periodo de ataque, la diferencia entre los Jain Index se va reduciendo, llegando al punto en el cual conseguimos la misma justicia para todas las estaciones, independientemente de la configuración del sistema (número de estaciones atacadas).

Cuando se incrementa el periodo de ataque, el número de estaciones y el tamaño de la trama, el punto de inflexión se va desplazando hasta el momento en el que desaparece. De esta forma se obtiene la misma justicia para todas las estaciones.

4.4. Estudio del consumo de baterías.

Mediante un modelo analítico y tomando de base el consumo del sistema cuando no existen ataques, vamos a estudiar el efecto que produce el ataque sobre el mismo.

4.4.1. Modelo analítico

En este modelo se distingue el consumo de las estaciones atacadas y el consumo de las estaciones no atacadas.

Se emplea la formula 4.10 para el cálculo del consumo de las estaciones atacadas.

$$C(mWh) = P\,(mW) \cdot T(h) \qquad \textbf{(4.10)}$$

Este consumo vendrá definido por el consumo del tiempo de ataque, comprendido entre los 60s de espera antes de la autenticación y la misma, descartando el tiempo de desautenticación por ser despreciable frente a los otros dos valores.

Por tanto y basándonos en la fórmula anterior (4.10), obtendremos el consumo referente al tiempo de espera producido por la desautenticación y el consumo de la autenticación.

$$C = 60s \cdot P_{libre} + C_{autenticación} \qquad \textbf{(4.11)}$$

Donde P_{libre} es la potencia cuando el canal está inactivo y $C_{autenticación}$ es el consumo durante la autenticación. Ambos valores son extraídos del proyecto final de carrera de Didac Mediavilla [3]. En la tabla 4.24 se muestras los valores

del consumo durante la autenticación y la tabla 4.25 presenta los valores de la potencia para los diferentes estados de la tarjeta.

Tabla 4.24 Consumo durante la autenticación (mWh).

	3Com Pc1	3Com Pc2	3Com Pc3	Intel Pc1	Atheros Pc2	Atheros Pc3	Cisco Pc1	Cisco Pc2
TLS	0,0693	0,0461	0,0354	0,0605	0,0395	0,0601	0,0597	0,0413
PEAP	0,0593	0,0585	0,0367	0,0518	0,0365	0,0640	0,0566	0,0369
TTLS	0,1701	0,1082	0,1103	0,1437	0,1205	0,1515	0,1287	0,1280
LEAP				0,0295	0,0137		0,0401	0,0145
LEAP (AP+Radius)				0,0167	0,0120		0,0314	0,0124

	PAP	CHAP	MSCHAP	MSCHAPV2
TTLS Software Intel)	0,000341	0,000342	0,000345	0,000374

Tabla 4.25 Potencia para los diferentes estados de cada tarjeta.

Marca	3Com	Intel	Atheros	Cisco
Potencia Tx [dBm]	16	16	17	20
Consumo Idle [mW]	750	60	660	669,9
Comsumo Tx [mW]	1000	1450	1419	1828,2
Consumo Rx [mW]	1000	850	1419	1049,4

Para el cálculo del consumo de las estaciones no atacadas nos basaremos en el consumo del número de tramas que se puede enviar durante el tiempo de recuperación de un ataque (60s + Tiempo de autenticación). El consumo de las STAs no atacadas viene determinado por los diferentes tiempos y consumos durante los estados de inactividad, transmisión y recepción que se dan en una transmisión con éxito.

Se evaluará el consumo de las STAs no atacadas para los diferentes protocolos de autenticación, variando el tamaño de la trama, el número de STAs y el periodo del ataque.

Mediante la fórmula 4.12 calculamos el consumo de las STAs no atacadas.

$$Consum = n^{\underline{o}}\ de\ tramas \cdot Consumo_{trama} \quad \textbf{(4.12)}$$

Donde el número de tramas vendrá calculado de la siguiente manera (4.13):

$$n^{\underline{o}} de\ tramas = \frac{60s + T_{autenticación}}{T_{trama}} \quad \textbf{(4.13)}$$

T_{trama} es el tiempo de trama que toma el valor siguiendo la fórmula (4.14):

$$T_{trama} = DIFS + Ttx_{datos} + SIFS + t_{ACK} + T_{backoff} \quad \textbf{(4.14)}$$

Donde Ttx_{datos} es calculado según la fórmula (4.15):

$$Ttx_{datos} = PLCP + 4\left\lfloor \frac{6+16+8\cdot(MAC\ header+FCS+Payload)}{4\cdot V_{tx}} \right\rfloor + Signal\ Extension \quad \textbf{(4.15)}$$

Donde PLCP es el preámbulo físico más la cabecera (20 μs), el signal extensión es 6 μs y la velocidad de transmisión es de 54 Mbps.

El t_{ACK} se calcula aplicando (4.15) con el campo *payload* de 14 bytes y MAC header de 0 bytes.

Para el cálculo de $T_{backoff}$ nos basamos en el artículo de Eduard García [11] donde se define TBO_i como el tiempo medio que tiene que esperar una estación *i* a que expire el temporizador de backoff. TBO_i depende del número de transmisiones previas. El valor medio del intervalo de backoff, $T_{BO}(j)$ después de j transmisiones consecutivas viene dado por la fórmula 4.16.

$$T_{BO}(j) = \begin{cases} \frac{2^j\,(CWmin+1)}{2}\,Tslot & 0 \le j \le 6 \\ \frac{CWmax}{2}\,Tslot & j \ge 6 \end{cases} \quad \textbf{(4.16)}$$

Se considera que el número de retransmisiones necesarias para una transmisión con éxito, es una variable geométrica distribuida aleatoriamente.

Si P_i se define como la probabilidad de que una trama enviada por *i* sea retransmitida, se define el tiempo medio de backoff TBO_i con la fórmula 4.17.

$$TBO_i = \sum_{j=0}^{\infty}(1 - P_i){P_i}^j \cdot T_{BO}(j) \quad \textbf{(4.17)}$$

Para calcular TBO_i y $T_{BO}(j)$ se programaron dos funciones que se explican en el anexo I.

Para calcular el $Consumo_{trama}$ (fórmula 4.18), cada tiempo de inactividad, transmisión y recepción, se multiplicará por la potencia de cada tarjeta (transmisión, recepción o libre). Estos valores de potencia los podemos consultar en la tabla 4.21.

$$Consumo_{trama} = DIFS \cdot Consumo_{libre} + Ttx_{datos} \cdot Consumo_{tx} + SIFS \cdot Consumo_{libre} + T_{ACK} \cdot Consumo_{rx} + T_{backoff} \cdot Consum_{libre} \quad \textbf{(4.18)}$$

En la tabla 4.26 se muestra el consumo de las estaciones atacadas y no atacadas para la tarjeta Intel y el pc1 con el protocolo de autenticación EAP-TLS. En el anexo J se muestran los resultados con el resto de protocolos de autenticación.

Tabla 4.26 Consumo de las baterías de las STAs atacadas y no atacadas

Número de estaciones	Tamaño de la trama	Consumo no atacadas	Consumo atacadas
2	200	14,441	12,561
4	200	14,370	12,561
10	200	14,234	12,561
15	200	14,158	12,561
20	200	14,099	12,561
50	200	13,850	12,561
100	200	13,603	12,561
2	600	14,970	12,561
4	600	14,899	12,561
10	600	14,760	12,561
15	600	14,680	12,561
20	600	14,617	12,561
50	600	14,340	12,561
100	600	14,048	12,561
2	1000	15,301	12,561
4	1000	15,235	12,561
10	1000	15,104	12,561
15	1000	15,026	12,561
20	1000	14,965	12,561
50	1000	14,688	12,561
100	1000	14,382	12,561
2	1500	15,574	12,561

4	1500	15,515	12,561
10	1500	15,396	12,561
15	1500	15,325	12,561
20	1500	15,268	12,561
50	1500	15,004	12,561
100	1500	14,699	12,561

Como se observa en la figura 4.43, el consumo de las STAs atacadas no cambia debido a que éstas no dependen ni del tamaño de la trama ni del número de estaciones. En el anexo J vemos como este valor varía según el escenario y el protocolo de autenticación, ya que éste se predetermina según el valor del consumo de autenticación. Por lo tanto, el consumo de las STAs atacadas será mayor cuanto más grande sea el consumo de autenticación.

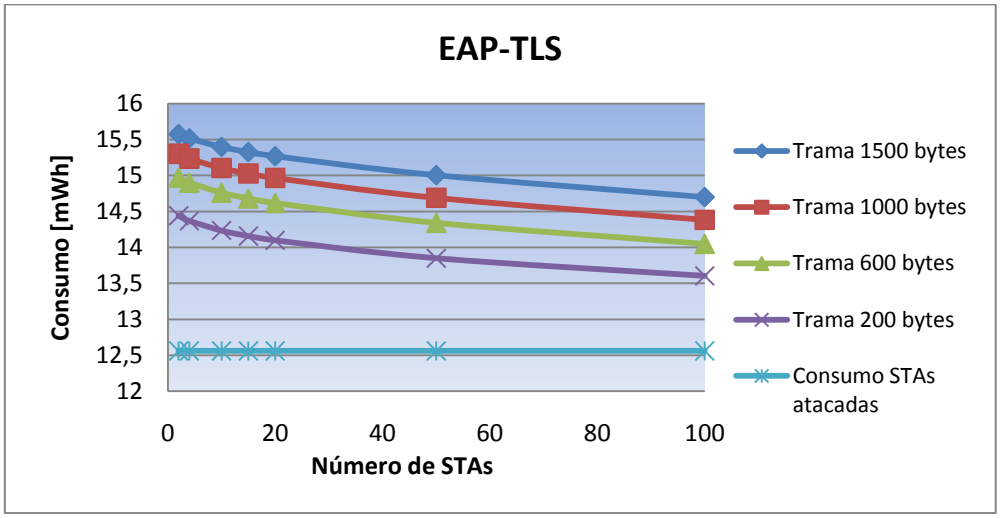


Fig. 4.43 Consumo del escenario Intel en pc1 empleando EAP-TLS

Para el consumo de las STAs atacadas, se alcanza el valor más elevado de 12,67 mWh, trabajando con la tarjeta 3Com en Pc1 y el protocolo de autenticación EAP-TTLS. Por otro lado, se consigue el menor consumo con un 12'512 mWh empleando la tarjeta Atheros en pc2 con el protocolo de autenticación EAP-LEAP (AP + servidor radius), siendo 1,24% la diferencia entre ellos.

Por otro lado, el consumo de las STAs no atacadas aumenta si aumentamos el tamaño de la trama de datos y disminuye con el incremento del número de STAs. Obtenemos el mayor consumo de 15,71 mWh utilizando el escenario 3Com en pc1 con el protocolo EAP-TTLS. Se obtiene el menor consumo utlizando la tarjeta Cisco en pc2 con el protocolo EAP-LEAP (AP+servidor RADIUS), obteniendo un consumo de 13,55 mWh, es decir, un 13,75% menos.

Con los datos obtenidos podemos concluir que el consumo de las STAs no atacadas es un 20,35% mayor que el de las STAs atacadas. Este evento

ocurre porque las STAs atacadas permanecen más tiempo inactivas como consecuencia del ataque, ahorrándose de esta forma el consumo de la transmisión.

4.4.2. Ejemplo sobre un caso práctico

Partimos de un escenario con la estación1, que es el ordenador portátil (3.1), que utiliza la tarjeta de red Intel Pro 2200 BG.

Podemos extrapolar estos datos a un modelo real gracias a los resultados obtenidos del trabajo final de carrera de Jaume Giribert y Miguel Madrid [4] donde se puede ver el consumo de la estación 1, vinculada a la red EDUROAM de la UPC. Dicho consumo se obtuvo mediante un analizador de potencia Agilent N6705A.

Por tanto, sabiendo de dicho trabajo que la batería de la estación1 tiene una energía de batería de 53,28 Wh, y mediante los datos obtenidos por nuestro modelo, se deduce la siguiente fórmula (4.19):

$$Tiempo\ restante\ bateria_{no\ atacadas} = \frac{energia\ de\ la\ batería\ [Wh]}{Potencia_{media}[w] + Potencia_{tarjeta}[w]} \quad \textbf{(4.19)}$$

Donde la potencia media se obtiene mediante los resultados [4], siendo de 25,563 W. De esta forma conseguimos un tiempo restante de batería de (4.20):

$$Tiempo\ restante\ bateria_{no\ atacadas} = \frac{53{,}28}{25{,}563 + 0{,}01571} = 2{,}082\ h \quad \textbf{(4.20)}$$

En el caso de las estaciones atacadas el tiempo restante será:

$$Tiempo\ restante\ bateria_{atacadas} = \frac{53{,}28}{25{,}563 + 0{,}01267} = 2{,}083\ h$$

$$2{,}083\ h - 2{,}082\ h = 0{,}001\ h$$

$$0{,}001\ h \cdot \frac{3600\ s}{1\ h} = 3{,}6\ s.$$

Así, haciendo la diferencia entre ambos tiempos, podemos ver que cuando las estaciones están atacadas, el tiempo restante de la batería aumenta en 3,6s en cada periodo de ataque. Si empleamos un periodo de ataque de 90s durante

un tiempo de simulación de 1800s, obtendremos una ganancia de batería de 72s.

$$número\ de\ ataques_{por_{simulación}} = \frac{1800\ s}{90s} = 20\ ataques.$$

$$20\ ataques \cdot 3{,}6\frac{s}{ataque} = 72\ s$$

CAPÍTULO 5. CONCLUSIONES

En este proyecto, se realiza un estudio de la seguridad de las redes WLAN evaluando las contramedidas que se emplean para combatir el ataque de falsificación. El hecho de aplicar estas contramedidas de forma reiterada, puede provocar DoS, afectando a factores como, el throughput, la justicia y el consumo de las baterías.

Así mismo, uno de los objetivos de este trabajo era verificar el efecto del ataque DoS sobre el throughput, la justicia y el consumo de las baterías.
En el estudio del throughput se distinguieron dos escenarios, el primero cuando se atacaba al AP y el segundo cuando se atacaban a las STAs.

Del primer escenario se estudió como afectaba el aumentar el periodo del ataque, el tamaño de la trama, el número de STAs y el tráfico ofrecido. Para ello se realizaron las pruebas con el protocolo TLS y se realizo una comparativa de las distintas tarjetas de red empleadas con los diferentes protocolos de autenticación estudiados en este trabajo. Se puede concluir que aumentando el periodo de ataque y el tamaño del paquete de datos, el throughput aumenta. De lo contrario, aumentando el número de estaciones el throughput disminuye.

Otra de las conclusiones es que la proporción entre los resultados del throughput para los distintos periodos y las diferentes tramas analizadas, se mantiene constante, independientemente del número de estaciones y del tamaño de la trama.

En el caso del aumento del tráfico ofrecido, se ha observado que el punto de saturación del throughput aumenta con el tamaño de la trama de datos, ya que al aumentar el tamaño se cursa más tráfico.

Para la comparativa de los distintas escenarios empleando los diferentes protocolos, se llegó a la conclusión que el valor máximo de throughput obtenido lo conseguimos con el protocolo LEAP (AP+ servidor RADIUS) y el escenario Atheros en pc2.

Una vez obtenidos los resultados mediante el simulador, se hizo una comparativa con un modelo analítico para verificar los resultados, observándose una diferencia de alrededor de un 10% para tramas de 200 bytes.

Partiendo del segundo escenario, donde las STAs están siendo atacadas y el AP no, se concluye, que si aumenta el número de estaciones, el throughput de las STAs atacadas y el throughput de las STAs no atacadas decrece.

A medida que se aumenta el número de estaciones atacadas, el throughput de las STAs atacadas permanece constante y a medida que aumentamos el periodo de ataque, éste incrementa. El throughput de las STAs no atacadas

disminuye conforme el periodo de ataque va aumentando, y crece con el número de STAs atacadas

En la evaluación de la justicia con la que se accede al medio se puede concluir que el incremento del periodo del ataque hace que el Jain Index aumente. A medida que se va aumentando el número de estaciones atacadas, la justicia disminuye hasta llegar a un punto donde los mecanismos de la red aplican sus mecanismos para evitar la monopolización de la red.

En referencia al estudio del consumo de baterías, mediante un modelo analítico y tomando de base el consumo del sistema cuando no existen ataques, se comprueba que el consumo de las STAs atacadas es menor que el consumo de las STAs no atacadas. Este evento ocurre porque las STAs atacadas permanecen más tiempo inactivas como consecuencia del ataque, ahorrándose de esta forma el consumo de la transmisión.

En el caso donde las STAs no están siendo atacadas, se alcanza el mayor consumo con un valor de 12,67 mWh, trabajando con el escenario 3Com en pc1 y el protocolo de autenticación EAP-TTLS. Por otro lado, se consigue el menor consumo con un 12’512 mWh empleando el escenario Atheros en pc2 con el protocolo de autenticación EAP-LEAP (AP + servidor radius), siendo 1,24% la diferencia entre ellos.

5.1 Implicaciones medioambientales

En todo estudio, se pueden evaluar los factores medioambientales implicados en el mismo, en el caso que nos atañe, partimos de un estudio mediante el uso del simulador 802.11, una aplicación de software, y el hardware sobre el que se ejecutan las simulaciones.

Al finalizar este trabajo, dada la naturaleza de simulación de sus pruebas, podemos decir que el impacto ambiental ha sido bajo, ya que no hemos empleado hardware físico para obtener los datos de potencia, sino el simulador y datos obtenidos de otros documentos.

En el apartado de consumo eléctrico, el hardware usado consistía en el PC portátil estación1 que ya estaba en nuestra posesión y ha sido usado para realizar otros trabajos de final de carrera, y los dos servidores, estación2 y estación3, que se encuentran emplazados en la sala de servidores de la fundación i2CAT, obteniendo el beneficio de la gestión del consumo de la sala, optimizando el uso de los mismos ya que además estos servidores son compartidos con otros proyectos.

Para las pruebas de consumo se han aprovechado los datos de proyectos anteriores con el consiguiente ahorro energético, al no haber tenido que repetir dichas pruebas que incluían el uso de medidores de potencia y otros dispositivos que habrían producido un mayor impacto.

Por tanto concluimos que el impacto medioambiental del trabajo ha sido moderado, ya que los ordenadores usados se encontraban en todo momento en uso para otros menesteres, no desaprovechándose su tiempo de operación, y limitando el tiempo de simulación al usar los tres ordenadores en paralelo para realizar las pruebas.

ABREVIACIONES

Nombre completo	Siglas y acrónimos
Acknowledgment	ACK
Advance Encryption Standard	AES
Access Point	AP
Basic Service Set	BSS
Clear Channel Assessment	CCA
CTR with CBC-MAC Protocol	CCMP
Challenge Handshake Authentication Protocol	CHAP
Cyclic redundancy check	CRC
Carrier sense multiple access with collision avoidance	CSMA/CA
Clear to Send	CTS
Contention Window	CW
Distribution Coordination Function	DCF
DCF Interframe Space	DIFS
Denial-of-service	DoS
Distributed System	DS
Direct-sequence spread spectrum	DSSS
Extensible Authentication Protocol	EAP
Extended InterFrame space	EIFS
Extended Basic Service Set	ESS
Flexible Authentication via Secure Tunneling	FAST
Frame Check Sequence	FCS
Frequency Hopping Spread Spectrum	FHSS
Hybrid Coordination Function	HCF
Independent Basic Service Set	IBSS
Integrity Check value	ICV
Identifier	ID
Institute of Electrical and Electronics Engineers	IEEE
Interframe spacing	IFS
Infrared radiation	IR
Initalization Vector	IV
Lightweight Extensible Authentication Protocol	LEAP
Media Access Control Address	MAC
Message Integrity Code	MIC
MAC Protocol Data Unit	MPDU
Microsoft CHAP	MSCHAP
MAC service Data Unit	MSDU
Network Allocation Vector	NAV
Orthogonal frequency-division multiplexing	OFDM

Pasword Authentication Protocol	PAP
Point Coordination Function	PCF
Protected Extensible Authentication Protocol	PEAP
PHYsical	PHY
Point Coordination IFS o Access Point Coordination IFS	PIFS
Physical Layer Convergence Procedure	PLCP
Packet Number	PN
Pre-shared key	PSK
Quality of service	QoS
Robust Security Network Association	RSNA
Request to Send	RTS
Short Interframe Space	SIFS
Service Set IDentifier	SSID
STAtion	STA
Target Beacon Transmission Time	TBTT
Temporal Key Integrity Protocol	TKIP
Transport Layer Security	TLS
TKIP Sequence Counter	TSC
Tunneled TLS	TTLS
Universitat Politècnica de Catalanya	UPC
Wired Equivalent Privacy	WEP
Wireless Local Area Network	WLAN
Wi-Fi Protected Access	WPA

BIBLIOGRAFÍA

[1] G. Bianchi, *Performance Analysis of the IEEE 802.11 Distributed Coordination Function*, IEEE Journal on selected *areas in communications*, Vol. 18, No. 3, pp. 535 – 547, 3Marzo 2000.

[2] E. López Aguilera*, Contributions to the evaluations and Enhancement of WLAN IEEE 802.11 Medium Access Control Mechanism*, Tesis Doctoral, Junio 2008.

[3] Mediavilla Urra, Dídac*. Seguridad en WLAN IEEE 802.11: Evaluación de los mecanismos de cifrado y autenticación.* PFC. Castelldefels: EPSC, 2 de abril 2009.

[4] Giribert Peraire, Jaume y Madrid Villar, Miguel. *Xarxa cel·lular de 4G basada en IPv6: Desenvolupament d'un demostrador (II).* TFC. Castelldefels: EPSC, 8 de mayo 2009.

[5] Mathew Gast, 802.11 Wireless Networks: The Definitive Guide. Ed.O'Reilly, 2005.

[6] Rufi, Antoon W, *Network Secuity: 1 and 2 Companion Guide.* Ed.Cisco Press 2007.

[7] Batalle Alcalde, Oriol. *Seguridad 802.11: Estudio y desarrollo de un sistema de gestión para EAP-TLS.* PFC. Escola tècnica superior d'enginyeria de telecomunicació de Barcelona. 6 de julio de 2009

[8] IEEE Standards Association. *802.11-2007. Part 11: Wireless LAN MAC and PHY specification.* [en línea]. Disponible en : http://standards.ieee.org.

[9] He, Changhua y C.Mitchell, John. *Analysis of the 802.11i 4-Way-Handshake.* Electronical Engineering and computer science department. Stanford University.

[10] Krishna S., Sundaralingam S., Darrin M., Balinsky.Hall, J.A., "Cisco Wireless LAN Security", Cap. 7 en EAP Authentication Protocols for WLANs Feb 18, 2005.

[11] García, Eduard. *Client- Driven load balancing in 802.11 WLANs.* European Transactions on Telecommunications Volume 20 Issue 5, 494 – 507.

[12] Krishna Sankar, Sri Sundaralingam, Darrin Miller and Andrew Balinsky. *Cisco Wireless LAN Security, Chapter 7: EAP Authentication Protocols for WLANs,* Editorial: Pearson Education**.**

[13] Web del IEEE del protocolo IEEE 802.1X. [En línea]Disponible en: http://www.ieee802.org/1/pages/802.1x-2004.html

[14] RFC EAP. [En línea]. Disponible en: http://www.ietf.org/rfc/rfc3748.txt.

[15] RFC EAP-TLS. [En línea] Disponible en: http://www.ietf.org/rfc/rfc5216.txt

[16] Draft PEAP. [En línea]. Disponible en: http://tools.ietf.org/html/draft-josefsson-pppext-eap-tls-eap-01.

[17] Draft EAP-TTLS.[En línea]. Disponible en: http://tools.ietf.org/html/draft-funk-eap-ttls-v0-05

Escola Politècnica Superior de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

ANEXOS

TÍTULO DEL TFC: Seguridad en WLAN 802.11: Evaluación de las contramedidas para combatir el ataque de falsificación

TITULACIÓN: Ingeniería Técnica de Telecomunicaciones, especialidad Telemática

AUTOR: Mireia García Dueñas

DIRECTOR: Elena López Aguilera

FECHA: 14-05-2010.

A. Trama Michael MIC Failure Report

Como se explica en el capítulo 2, el protocolo TKIP envía un Michael MIC failure Report mediante una EAPoL key frame indicando el fallo en la MIC. Para indicar que lo que transporta es una EAPoL key frame, el campo EAP Code tiene como valor 4 (véase figura B1.)

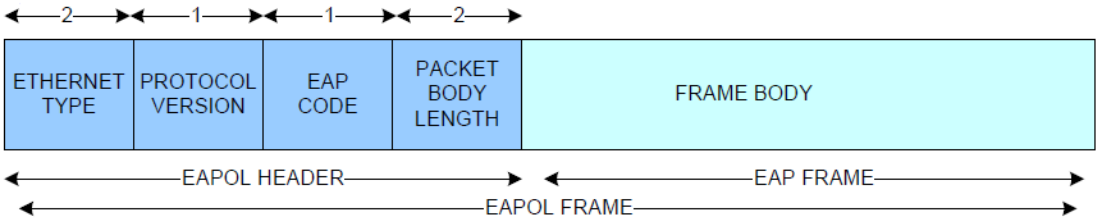


Fig. B1 Trama EAPoL

Dentro del frame body viaja la EAPoL key frame con el formato de la figura B2. Para indicar que es un MIC failure Report los bits del campo Key Information mostrados en la figura B2 deben estar a 1: MIC bit, Error bit, Request bit y Secure bit.

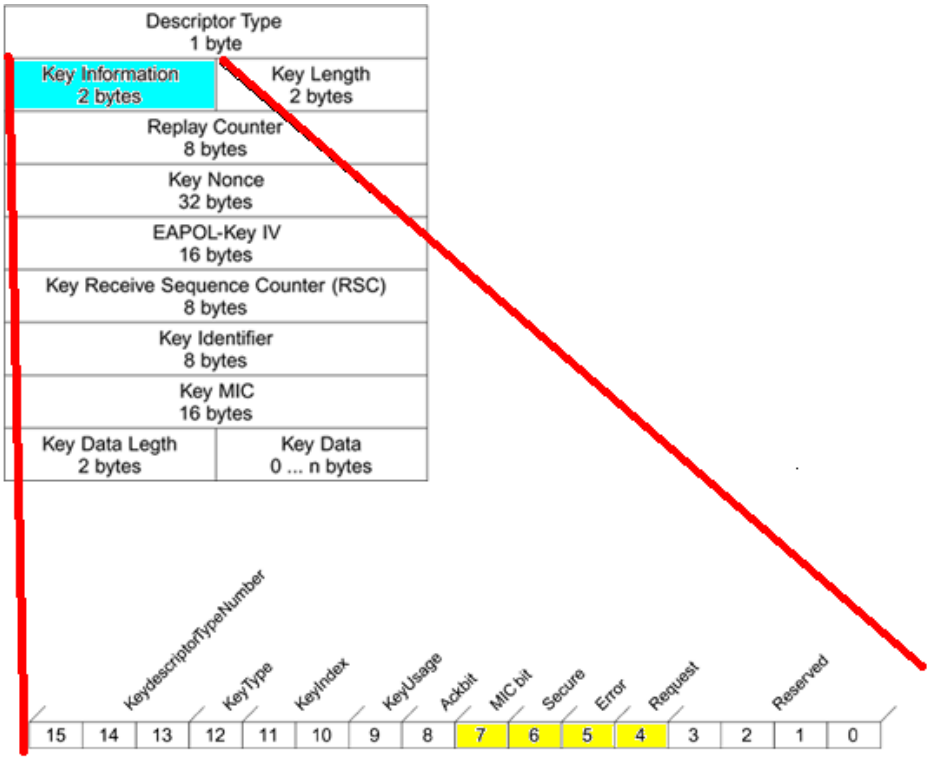


Figura B2. Trama EAPoL key frame.

B. Manual del simulador 802.11

El simulador 802.11 es una aplicación que funciona sobre el programa Borland C++ Builder 3 cuya función es simular y analizar un escenario 802.11.

Para cada escenario se crea una carpeta del tipo Sim X, donde X será un número que identificará el escenario. Dentro de cada carpeta se introducirán los archivos de entrada a partir de los cuales se generará una carpeta de archivos resultantes, en esta última podremos observar los resultados obtenidos de nuestra simulación.

En la figura B1 podremos ver un esquema en el que podremos entenderlo mejor:

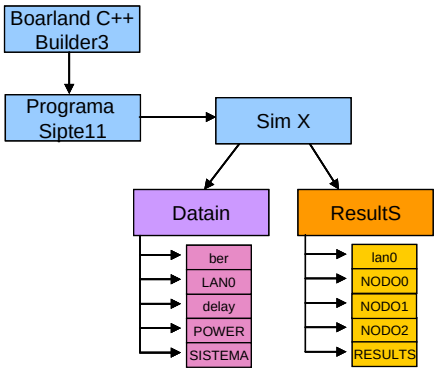


Fig. B1 UML del simulador 802.11

B.1. Un poco de C++ Bulider

C++ Builder es una aplicación Windows que proporciona un entorno de trabajo visual para construir aplicaciones Windows que integra distintos aspectos de la programación en un entorno unificado o integrado.

B.1.1. Una visión general del C++ Builder.

El entorno de desarrollo se divide, básicamente, en tres partes. Una serie de ventanas, que pueden estar visibles u ocultas, constituyen la base de C++ Builder. El aspecto de la aplicación al inicio de una sesión de trabajo es el mostrado en la figura B2.

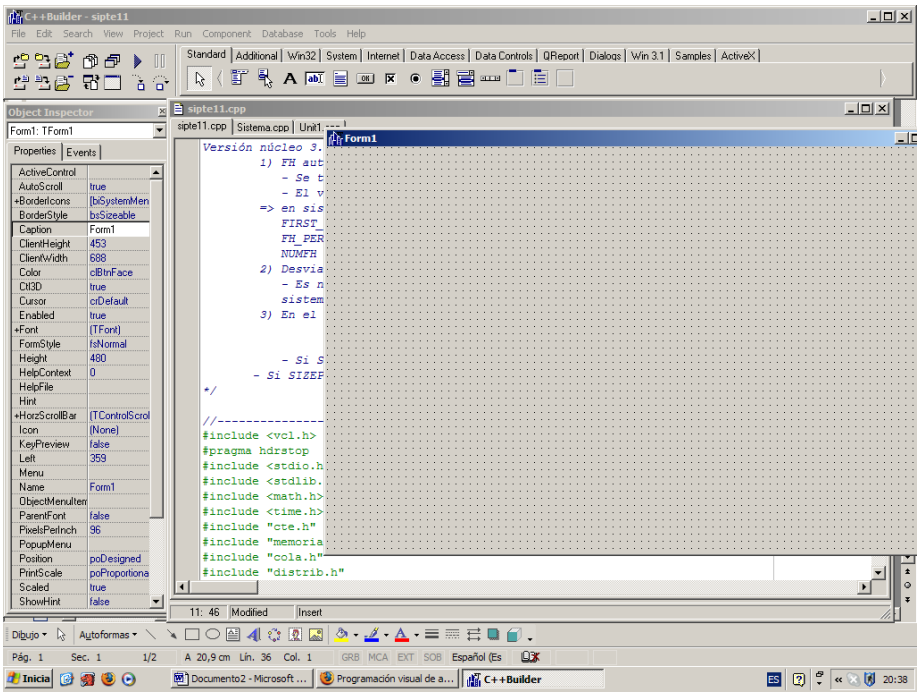


Fig.B2. Aspecto del C++ Builder al inicio de la sesión

En la parte superior se coloca la ventana principal, que contiene el menú principal, la barra de herramientas (a la izquierda) y la paleta de componentes (a la derecha). Debajo de la ventana principal, y a la izquierda se coloca el inspector de objetos. A la derecha del inspector de objetos está el área de trabajo de C++ Builder, que inicialmente muestra el diseñador de formularios, y escondido u oculto parcialmente tras éste aparece el editor de código.

- **Ventana principal.**
  En la ventana principal se ubican el menú principal, la barra de herramientas y la paleta de componentes (figura B3).

.

Fig.B3 Ventana principal de C++ Builder

- **Menú principal**: Permite el acceso a todas las operaciones y posibilita la configuración del programa.

- **Barra de herramientas**: Permite un acceso rápido a las operaciones que se realizan más frecuentemente.

- **Paleta de componentes**: Agrupa a los componentes que pueden incluirse en las aplicaciones.

- **Inspector de objetos**: Para cambiar las *propiedades* de los objetos que forman la aplicación y seleccionar los *eventos* a los que debe responder la aplicación. (véase figura B4).

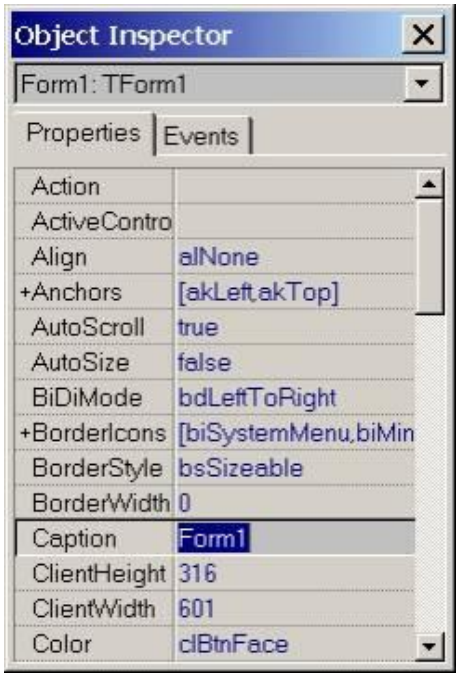


Fig.B4 El inspector de objetos.

- **Diseñador de formularios**: Es una ventana cuadriculada como la mostrada en la figura C5, sobre el que se disponen los componentes para diseñar las ventanas que formarán la aplicación.

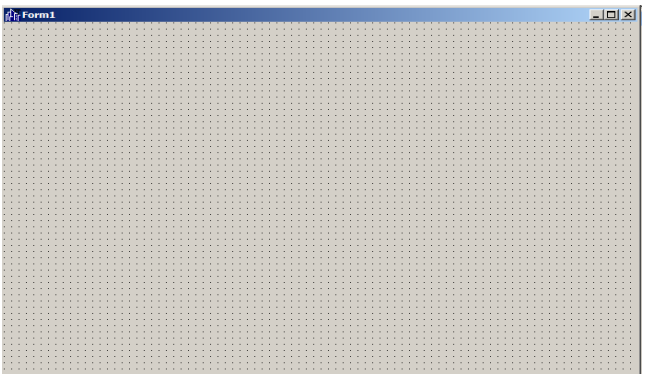


Fig.B5. El diseñador de formularios.

- **Editor de código:** típico editor de texto multiventana para ver y editar el código de la aplicación. Está perfectamente integrado con el inspector de objetos y el diseñador de formularios (figura B6).

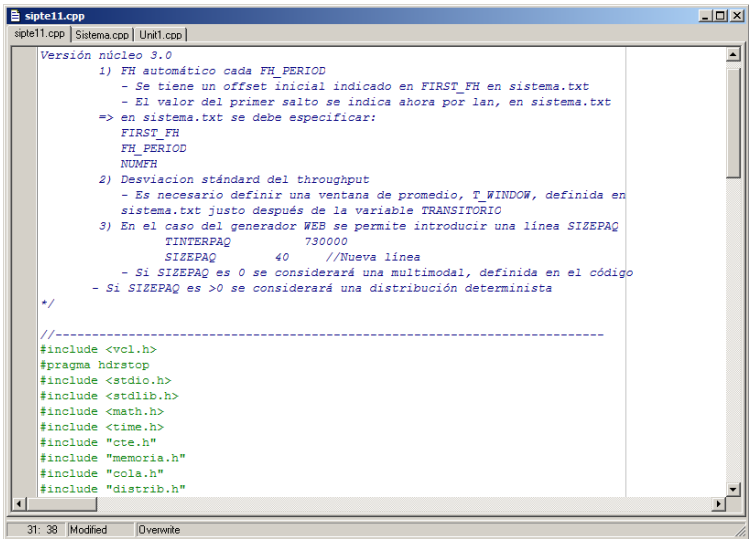


Fig.B6. El editor de código.

En la ventana del editor pueden "pegarse" el gestor de proyectos y el inspector de clases aunque estas dos herramientas pueden aparecer también como ventanas separadas.

- **Inspector de clases**: Es un navegador que muestra las clases, objetos y métodos asociados a la aplicación. Aparece por defecto asociada al editor. Para abrir esta ventana: View → ClassExplorer.

- **Administrador de proyectos**: Es básicamente un navegador entre los diferentes ficheros que forman la aplicación. No aparece por defecto, y cuando se abre (View → Project Manager) se muestra como una ventana independiente.

  - **Administrador de proyectos**

  Un proyecto es un conjunto de archivos que trabajan en equipo para crear un archivo ejecutable independiente o una DLL. Un grupo de proyectos es un conjunto de proyectos.

  Los proyectos que componen un grupo de proyectos, y los archivos que componen cada uno de esos proyectos, es lo que presenta, en forma de árbol, el administrador de proyectos. Puede emplearse como navegador para seleccionar el módulo con el que se va a trabajar. Para visualizar el gestor de proyectos, seleccionar View → Project Manager.

  En todo momento existe un único proyecto activo (en la figura B7), y será este el que se ejecute si elegimos la opción Run →Run.

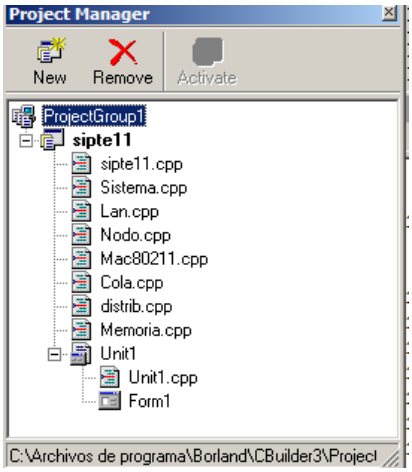


Fig.B7. El administrador de proyectos.

Los ficheros de proyecto especifican todos los recursos necesarios (ficheros .cpp, .h, ficheros de descripción de formularios, etc.) que se necesitan para la construcción del ejecutable. Los ficheros de proyecto tienen extensión .bpr y el ejecutable que se genera tiene el mismo nombre que el proyecto y la extensión .exe, lógicamente.

Todo proyecto en C++ Builder se compone, al menos, de un archivo de código que contiene la función principal (WinMain()). Su nombre es el nombre del proyecto, con la extensión .cpp (en la Figura B7). Este fichero no está, habitualmente, visible, ya que no es necesario modificarlo. Puede abrirse en el editor de código con la opción Project→View Source.

Cualquier aplicación típica tendrá al menos una ventana. Para cada ventana (en la FiguraB7, Form1) habrá un módulo, formado por una pareja de ficheros: un .cpp (en la FiguraB7) y su correspondiente .h: en el fichero .cpp estarán los gestores de los eventos asociados a los componentes de esa ventana y en el .h (que no se modificará, normalmente) estará la declaración de los componentes de esa ventana.

Además del fichero que contiene la función principal, un proyecto puede tener asociados una serie de módulos adicionales en los cuales pueden incluirse funciones y clases de objetos, como en cualquier aplicación C++. Cada uno de estos módulos estará formado por una pareja de ficheros: un .cpp y su correspondiente .h.

A un grupo de proyectos se le pueden añadir proyectos, archivos, formularios, módulos, nuevos o que ya existan.

En definitiva, el administrador de proyectos es únicamente un organizador de archivos. Veamos brevemente mediante la tabla B.1, qué tipos de archivos pueden formar parte de un proyecto/grupo de proyectos, y que cuál es su cometido:

Tabla B1. Tipos de archivos que pueden aparecer en un proyecto.

EXT	Descripción
BPR	Es el archivo makefile del proyecto. Define qué y cómo se debe compilar.
CPP	Archivos fuente de C++.
H	Archivos de cabecera de C++.
OBJ	Archivos objeto resultado de la compilación.
EXE	Es el programa ejecutable resultante.
TDS	Archivos temporales para la compilación incremental.
DFM	Archivos de descripción de formulario. Contiene los valores de las propiedades de cada componente. Aunque está en formato binario, puede verse como texto seleccionando View as text en el menú contextual que aparece al pulsar con el botón derecho del ratón cuando se está situado sobre el formulario.
RES	Un archivo de recursos.
DSK	Es el archivo que contiene la configuración del escritorio para un proyecto.
BPG	Es un archivo de grupo de proyectos. Describe qué proyectos conforman el grupo de proyectos.
HPP	Archivos de cabecera creados automáticamente por *C++ Builder*.

B.2. Gestor de ayuda

El sistema de ayuda será una de las herramientas que más útiles nos resultará en nuestro trabajo con C++ Builder. Especialmente la documentación de los componentes y clases predefinidas.

La ayuda es una ayuda estándar de Windows por lo que no entraremos en más detalles, sólo comentar que pulsando F1 obtendremos una ayuda contextual. (Figura B8).

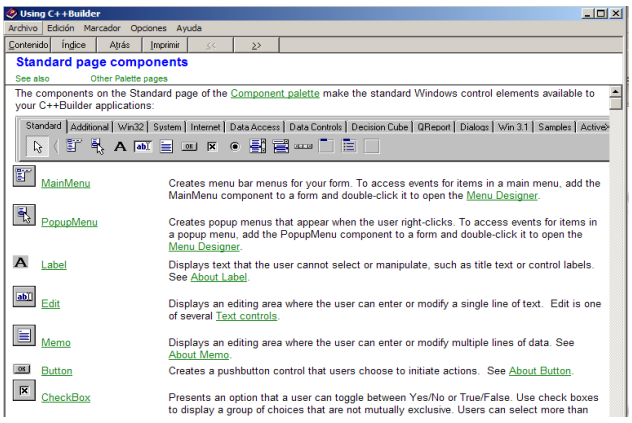


Fig.B8 Ventana que se muestra al apretar la tecla F1

Así, por ejemplo, en el caso de hacerlo en el editor de código, se nos ofrecerá la ayuda correspondiente a la palabra que se encuentre bajo el cursor.(figura B9)

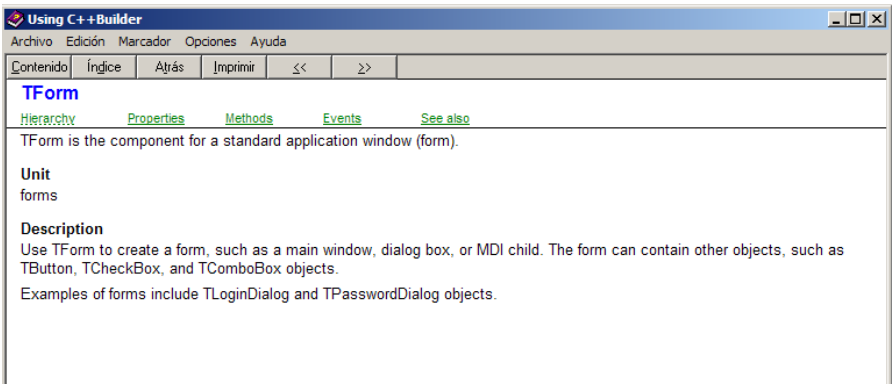


Fig.B9 Ayuda que se abre al pulsar F1 sobre la palabra clave T `Form`.

Pulsando F1 sobre cualquier opción de un menú se mostrará la ayuda asociada a esa opción.

B.3. Operaciones disponibles

Se puede decir que el menú principal es la ventana principal del IDE de C++ Builder y siempre está visible. En él podemos encontrar todas las operaciones disponibles.

Fig.B10. El menú principal.

En el título del menú principal aparece el nombre de la aplicación (C++ Builder) y el nombre del proyecto/grupo de proyectos con el que actualmente se está trabajando. Por defecto, asigna el nombre Project1 al proyecto con el que se va a trabajar. Como veremos, el concepto de proyecto es fundamental en C++ Builder ya que es el mecanismo de organizar sensatamente todos los ficheros (formularios, código fuente, recursos, etc.) asociados a una aplicación.
En nuestro caso describimos las principales.

- **Abrir el programa**

Primero de todo debemos saber cómo ejecutar el programa C++Builder. Para poder abrir el programa seguiremos la siguiente ruta:
Inicio → todos los programas → Borland C++ Builder 3→ C++Builder3.

- **Abrir un proyecto/fichero existente**

Nos situamos en el menú principal, en la pestaña File y seleccionamos la opción Open o Open Project. Si seleccionamos la opción Open nos saldrá una ventana donde por defecto nos mostrará los archivos (*.cpp, *.bpg, *.bpr, *.bpk), en cambio si escogemos la opción Open Project nos mostrará por defecto los archivos (*.bpg, *.bpr, *.bpk).

Otra Opión que tenemos es la de Reopen que nos facilita un listado de los últimos archivos abiertos/modificados.

- **Crear un proyecto**

En la pestaña File seleccionamos la opción New, como se muestra en la figura B11, contiene formularios, cuadros de diálogo, módulos de datos, asistentes, DLLs, etc. que podemos utilizar para simplificar el desarrollo de aplicaciones. Todos ellos están prediseñados y pueden servirnos como punto de partida para nuestros propios diseños. Además se pueden incorporar nuevos elementos que nosotros desarrollemos, consiguiendo de esta forma reutilizar nuestros diseños.

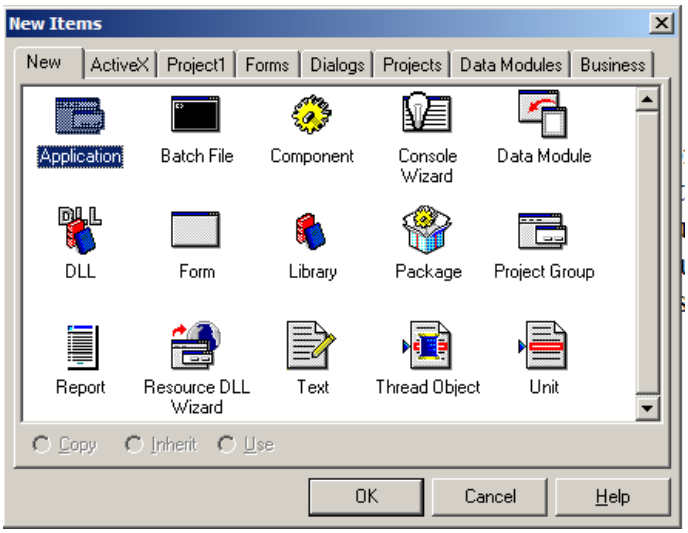


Fig. B11. Ventana New→File

- **Compilación, ejecución y depuración de programas.**

Nuestro objetivo es la creación de un programa ejecutable que se construye tomando como referencia los ficheros que forman el proyecto activo. Para esta tarea se utilizan los menús Project y Run.

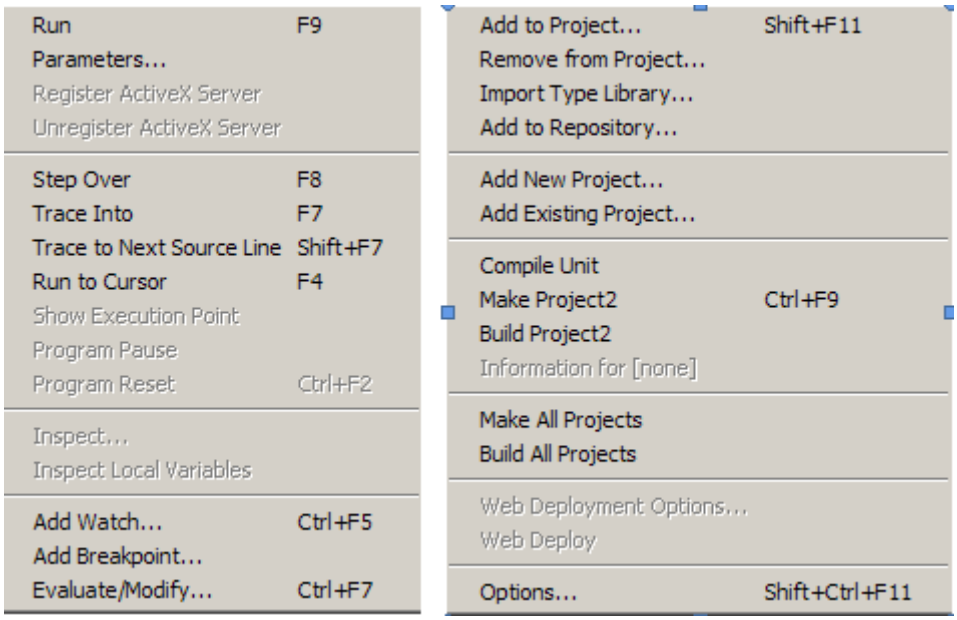


Fig.B12 Submenús Project y Run del menú principal.

En la compilación se trata de la obtención del programa ejecutable (extensión .EXE). Las operaciones asociadas a este objetivo se encuentran en el menú que se despliega al seleccionar la opción Project del menú principal, y las más importantes son:

- **Compile Unit:** Compila el modulo fuente activo (extensión `.CPP`) generando un fichero objeto (extensión `.OBJ`).
- **Make**: Genera el fichero ejecutable a partir de los ficheros objeto asociados al proyecto recompilando únicamente los módulos fuente que se hayan modificado desde la última vez que se compilaron.
- **Build**: Genera el fichero ejecutable a partir de los ficheros objeto asociados al proyecto recompilando todos los módulos fuente, aunque no se hayan modificado desde la última vez que se compilaron.

Si durante la compilación se detectaran errores, se mostraran en el editor de código y se puede acceder directamente a las líneas de código en las que se han detectado para facilitar su corrección.

Para ejecutar el programa basta con pinchar sobre el botón correspondiente de la barra de herramientas o seleccionar la opción Run → Run.

C++ Builder proporciona facilidades para la depuración de programas, seleccionables al desplegar el menú Run. Las más importantes son:

- **Step Over F8:** Ejecuta instrucción a instrucción el programa, pero ejecuta las llamadas a funciones como una instrucción más, sin mostrar la ejecución de las instrucciones de las funciones.

- **Trace Into F7:** Ejecuta instrucción a instrucción el programa, incluidas las instrucciones de las funciones llamadas.

- **Run to Cursor F4**. Ejecuta desde el principio del programa hasta la línea en la que está situada el cursor.

Para que la depuración sea más completa y versátil se incorporan las siguientes opciones en el mismo menú:

- **Add Watch Ctrl+F5**: Visualizar el contenido de una variable permanentemente.

- **Add Breakpoint**: Añade un punto de ruptura en la línea en la que está situado el cursor. de modo que cuando se ejecute el programa se detendrá su ejecución al llegar al siguiente punto de ruptura.

- **Guardar y salir**

Estas opciones las encontraremos en el menú File. Si queremos guardar escogeremos la opción Save y si queremos salir escogeremos la opción Close.

B.4. El simulador 802.11

Como hemos comentado con anterioridad, para cada escenario se crea una carpeta del tipo Sim X. Dentro de esta carpeta se encuentran las carpetas Datain y Results.

B.4.1. Archivos Datain

La carpeta Datain contiene los ficheros entrantes ber, delay, LAN0, power y sistem.

B.4.1.1. BER

Fichero donde se muestra una matriz de los resultados del BER en función de la C/I (relación entre la portadora y el ruido) y la modulación. (Véase figura B13).

```
//BER en función de la C/I(lineal) y de la modulación

//C/I       64-QAM      64-QAM      16-QAM      16-QAM      QPSK        QPSK        BPSK
//      v=54        v=48        v=36        v=24        v=18        v=12        v=9     v=6

C/I0        1       1       1       1       1       1       1       1
C/I1        1       1       1       1       1       1       1       0.188924
C/I2        1       1       1       1       1       0.123825    0.172769    0.000012
C/I3        1       1       1       1       1       0.000454    0.000376    0
C/I4        1       1       1       1       0.161041    0.000011    0.000006    0
C/I5        1       1       1       1       0.005642    0.000001    0       0
C/I6        1       1       1       1       0.000371    0       0       0
C/I7        1       1       1       0.254507    0.000042    0       0       0
C/I8        1       1       1       0.067267    0.000006    0       0       0
C/I9        1       1       1       0.01897     0.000001    0       0       0
C/I10       1       1       1       0.005809    0       0       0       0
C/I11       1       1       1       0.001958    0       0       0       0
C/I12       1       1       1       0.00073     0       0       0       0
C/I13       1       1       1       0.000299    0       0       0       0
C/I14       1       1       1       0.000133    0       0       0       0
C/I15       1       1       1       0.000062    0       0       0       0
C/I16       1       1       0.484266    0.000031    0       0       0       0
C/I17       1       1       0.23126     0.000016    0       0       0       0
```

Fig. B13 Fichero BER

B.4.1.2. POWER

Fichero de potencias (en dBm), donde las veremos representadas en una matriz según el número de AP y usuarios que tengamos en nuestra red.(Véase figura B14).

```
//Fichero de potencias
//Num Access Points 1
//Num usuarios 10
//  au11    su11_1  su11_2
au11    0   -55.0   -55.0
su11_1  -55.0   0.000000    -65.0
su11_2  -55.0   -65.0   0.000000
```

Fig. B14 Fichero POWER.

B.4.1.3. DELAY

Siguiendo la misma estructura que POWER.txt, en este fichero vemos una matriz donde se representa los retardos (µs) dependiendo del número de AP y usuarios que tenga nuestra red.

```
//Fichero de retardos
//Num Access Points 1
//Num usuarios 10
//  au11    su11_1  su11_2
au11    0   0.5 0.5
su11_1  0.5 0.000000    1.0
su11_2  0.5     1.0 0.000000
```

Fig.B15 Fichero DELAY

B.4.1.4. LAN0

Fichero donde se muestran los datos de cada uno de los nodos que componen la LAN. Hay que tener en cuenta que el AP será un nodo, por lo tanto, el número de nodos que contendrá la LAN será igual a al número de usuarios más el AP.

```
//Datos de cada uno de los nodos que componen la LAN
//NODO 0
IDNODO          0
TRACE           0
TIPO_NODO       AU
MBPS            0 54 54
GEAR_SHIFT      0 0 0
POLLABLE        0 0 0
MAX_REP_SHORT   10
MAX_REP_LONG    10
NUM_TU          0

//NODO 1
IDNODO          1
TRACE           0
TIPO_NODO       SU
MBPS            54
GEAR_SHIFT      0
POLLABLE        0
MAX_REP_SHORT   10
MAX_REP_LONG    10
NODOAU          0
NUM_TU          1

        //TU 0
        DESTINO         0
        TRAFICO         DCF
        TIPO            LEASELINE
        TINTERPAQ       494
        SIZEPAQ         1500

//NODO 2
IDNODO          2
TRACE           0
TIPO_NODO       SU
MBPS            54
GEAR_SHIFT      0
POLLABLE        0
MAX_REP_SHORT   10
MAX_REP_LONG    10
NODOAU          0
NUM_TU          1

        //TU 0
        DESTINO         0
        TRAFICO         DCF
        TIPO            LEASELINE
        TINTERPAQ       494
        SIZEPAQ         1500
```

Fig. B16 Fichero LAN0.

Cada nodo viene representado con las siguientes variables:

- **IDNODO:** Número con el que se identifica al nodo.
- **TIPO_NODO :** Variable que indica el tipo de nodo. Si la variable es AU→será AP, si la variable es SU→se trata de un usuario.
- **MBPS:** Vector que representa la velocidad utilizada entre el AP y el usuario.
- **0 54 54→** En este ejemplo, la matriz nos muestra que la red consta de un AP y de dos usuarios. El primer número es 0 ya que se trata de la relación AP-AP**,** el segundo número nos indica que la velocidad que hay

entre el AP y el primer usuario es 54 Mbps y el tercer número es el mismo que el segundo pero en este caso es la velocidad entre el AP y el usuario numero 2.

- **POLLABLE:** Matriz que indica si es sondeable**,** es decir, indica que es PCF.

- **MAX_REP_SHORT:** Número máximo de retransmisiones de tramas cortas (RTS,CTS).

- **MAX_REP_LONG:** Número máximo de retransmisiones de tramas largas (datos).

- **NODO AU:** Información que aparece en caso de que el tipo de nodo sea SU, nos indica a que AU está asociado.

- **DESTINO:** ID del AP al que se dirige**.**

- **TRAFICO:** Puede ser DFC o PFC.

- **TIPO**: indica el tipo de tráfico, en este caso, LEASELINE, indica tipo M/D/1.

- **TINTERPAQ:** Tiempo entre paquetes (microsegundos).

- **SIZEPAQ:** Tamaño del paquete (bytes)

B.4.1.5. SISTEMA

Fichero en el que encontramos los datos de simulación. (Figura B17).

```
//Datos de la simulación
SEMILLA          888
TSIMULACION      1800000000.000000
T_WINDOW         12000

//Datos del sistema
NUM_LAN          1
MAX_TU           106
PROB_TRACE       0.000000
MAX_ADJ          0
ADJ              0
MIN_NOISE        -96
UMBRAL_POTENCIA -86
MAX_HOPS         1

//Patrón de frecuencias, ftx frx del AU
ftx              1
frx              1

//Datos de la LAN 0
IDEMLAN          0
NUM_NODOS        3
SIZE_QUEUE       1000000000
FIRST_BACKOFF    0
UMBRAL_PAQ_AU    1000000000
UMBRAL_PAQ_SU    1000000000
SIZE_RTS         20
SIZE_CTS         14
SIZE_ACK         14
HEADER_DATA      34
PLCP             20
SignalExtension 6
MAX_DELAY        2.667358
SLOT_TIME        9.000000
SIFS             10.000000
PIFS             19.000000
DIFS             28.000000 28.000000
vmin             54.000000
EIFS             0
RX_TIMEOUT       7.667358
Cwmin            15 15
Cwmax            1023 1023
PF               2 2
GEAR_SHIFT       8 5 128
MIN_MBPS_CONT    0
SIZE_CF          400 //tamaño beacon
SIZE_CFEND       20
BEACON_PERIOD    1800000000.000000  //periodo beacon micros
FIRST_BEACON     1800000000.000000        // no poner 0.0
MAX_CFP          0.000000
CFP_PERIOD       500
DWELL_PERIOD     -1 -1 -1
FH_PERIOD        8000000000000.000000
```

Fig. B17. Fichero SISTEMA.

Se compone de las siguientes variables:

- **SEMILLA:** semilla de números aleatorios.
- **TSIMULACION** : tiempo de simulación.
- **T_WINDOW**: tiempo de ventana.
- **NUM_LAN** : Número de LANs.
- **MAX_ADJ**: número máximo de canales adyacentes.
- **ADJ:** Indica la potencia con la que se recibe la señal por el canal adyacente.
- **MIN_NOISE** : Ruido mínimo(dBm).
- **UMBRAL_POTENCIA**: Umbral de Potencia del mecanismo CCA.

- **MAX_HOPS**: número máximo de saltos en FHSS.
- **ftx** : frecuencia de transmisión.
- **frx**: frecuencia de recepción.
- **IDEMLAN**: Id de LAN.
- **NUM_NODOS**: Numero de nodos de LAN.
- **SIZE_QUEUE**: tamaño de la cola.
- **UMBRAL_PAQ_AU**: Umbral de paquete AP, de tal manera que si excede de este umbral, dejamos de tx deforma básica.
- **UMBRAL_PAQ_SU**: Umbral de paquetes de los usuarios o estaciones.
- **SIZE_RTS**: tamaño del paquete RTS.
- **SIZE_CTS** : tamaño del paquete CTS.
- **SIZE_ACK**. Tamañao del paquete ACK.
- **HEADER_DATA**: tamaño (bytes) de la cabecera de datos.
- **PLCP**: tiempo de transmisión del PLCP.
- **Signal Extension**: tiempo de transmisión de Signal Extensión.
- **MAX_DELAY**: Retardo máximo entre los usuarios de la LAN.
- **SLOT_TIME**: tiempo de Slot.
- **SIFS:** tiempo SIFS.
- **PIFS:** tiempo DIFS.
- **DIFS**: tiempo DIFS.
- **Vmin**: velocidad mínima utilizada en el sistema.
- **EIFS**: Extended interframe space.
- **RX_TIMEOUT**: timeout de recepción.
- **CWmin:** Ventana mínima de contienda
- **CWmax**: Ventana máxima de contienda.

- **PF**: Factor de persistencia.
- **MIN_MBPS_CONT**: indica si las tramas de control se envían a la Vmin. o no.
- **SIZE_CF**: tamaño del beacon.
- **SIZE_CFEND** : tamaño del paquete CFEND.
- **BEACON_PERIOD**: periodo beacon.
- **FIRST_BEACON**: tiempo del primer beacon.
- **MAX_CFP** tiempo máximo del CFP a este tiempo es al que se iguale el NAV.
- **CFP_PERIOD** : Periodo libre de contienda.

B.4.2. Archivos RESULTS

Los ficheros que se generarán al ejecutar el simulador serán los siguientes: *Results.txt, lan X.txt* , *NODO X.txt,fj_index.txt y lan_X.txt.*

Cuando decimos LAN X o NODO X, la X nos indica el ID de la LAN o del nodo, respectivamente, y dependerá del número de LANs y del número de nodos que tenga nuestro escenario. En nuestro escenario sólo hay una LAN que está formada por un AP y dos usuarios. Si ejecutamos el simulador obtendremos como resultado los siguientes ficheros: *Results.txt, lan0.txt, NOD0.txt* (AP)*, NODO1.txt* (Usuario1), *NODO2.txt* (Usuario2).

B.4.2.1. Lan0

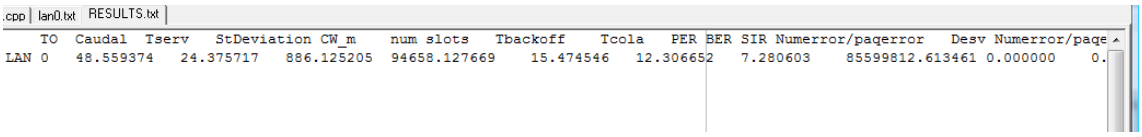
Fichero donde se muestra la información resultante de las LANs y cada uno de sus nodos.

```
//LAN 0

TO=                 48.5594
CaudalMbps=         24.3757
CaudalMbps AP=      0
CaudalMbps SU=      12.1879
CaudalMbpsDCF=      23.1965
TO beacon=          16.29
Retardo MAC=        845.625
    StD=        94658.1
Tserv=              886.125
    StD=        94658.1
Tserv AP=           +INF
    StD=        0
Tserv SU=           886.125
    StD=        94658.1
Tbackoff=       7.2806
Tbackoff AP=        0
Tbackoff SU=        7.2806
Tcola=                  8.55998e+07
Tcola AP=               +INF
Tcola SU=               8.55998e+07
Tcolision=              0.850669
PER=                0
PER Datos=          0
PER ACK=            0
SIR=                41
#col/paqok=         0.0291831
#col/paqok AP=  +INF
#col/paqok SU=  0.0291831
Pcolisión=          0.0283556
Pcolisión AP=       +INF
Pcolisión SU=       0.0283556
P col beacon=       0
#error/paqok=       0
#error ack=         0
#rtx/paqok=         +INF
#rtx exceso=        +INF

////////////////////////////////////////////
//NODO 0

TO_nodo=            0
Throughput=         0
Retardo mac=        +INF
    StD=        0
```

Fig. B18 Fichero lan0

B.4.2.2. Results

Fichero donde se muestra solamente los datos resultantes de las LAN.(Figura B19).



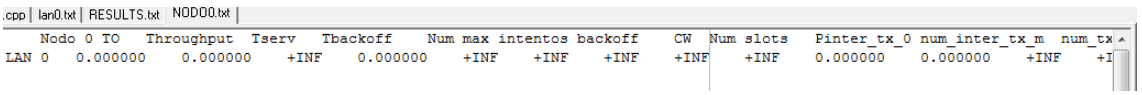
.cpp | lan0.txt | RESULTS.txt

	TO	Caudal	Tserv	StDeviation	CW_m	num slots	Tbackoff	Tcola	PER	BER	SIR	Numerror/paqerror	Desv Numerror/paqe
LAN	0	48.559374	24.375717	886.125205	94658.127669	15.474546	12.306652	7.280603	85599812.613461	0.000000	0.		

Fig. B19 Fichero Results.

B.4.2.3. NODO 0

Muestra solamente los datos resultantes del nodo 0. (Figura B20)



.cpp | lan0.txt | RESULTS.txt | NODO0.txt

	Nodo 0 TO	Throughput	Tserv	Tbackoff	Num max intentos backoff	CW	Num slots	Pinter_tx_0	num_inter_tx_m	num_tx
LAN	0	0.000000	0.000000	+INF	0.000000	+INF	+INF	+INF	+INF	+INF	0.000000	0.000000	+INF	+I

B.20 Fichero NODO 0.

B.4.2.4. NODO 1

Muestra solamente los datos resultantes del nodo. (Figura B21).

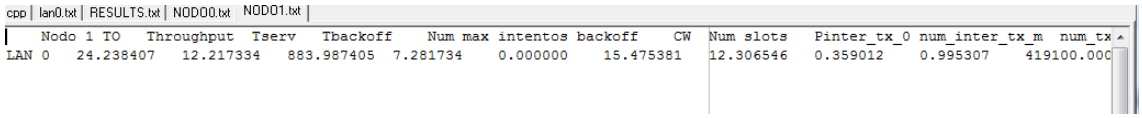


Fig. B21 Fichero NODO 1.

B.4.2.5. NODO 2

Muestra solamente los datos resultantes del nodo 2 (Figura B22)

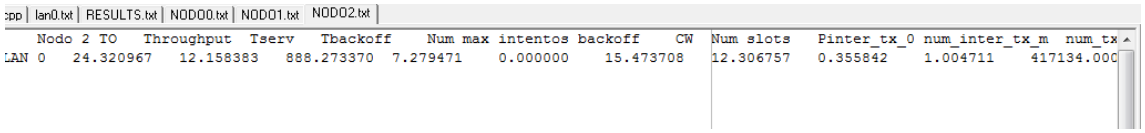


Fig B22. Fichero NODO 2.

B.4.2.6. Jain index

Muestra el índice de justica de la LAN. (Figura B23).

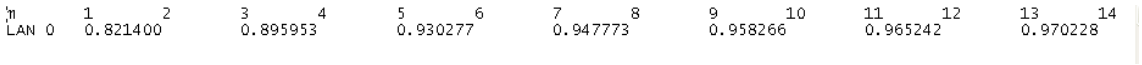


Fig. B23 Fichero Jain Index

B.4.2.7. Lan_X

Fichero donde muestra las veces que ha transmitido una estación mediante el identificador del nodo. (Figura B24).

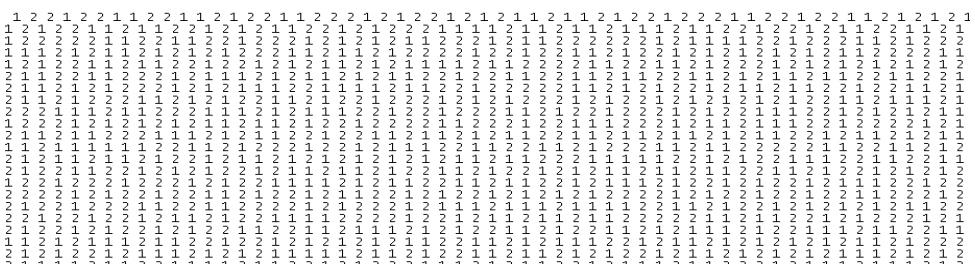


Fig. B24 Fichero Lan_X

C. Tabla de latencias

En este anexo se describe de forma más detallada las diferentes latencias empleadas para definir los tiempos de ataque.

El tiempo de ataque toma el valor de la suma de los 60 segundos (tiempo que tiene que esperar una STA para volver a autenticarse), del retardo de la desautenticación y el retardo de autenticación, el cual es medido a partir de los escenarios reales descritos anteriormente en las tablas 4.2 y 4.3.

Los valores del tiempo de autenticación son extraídos del proyecto final de carrera [3] y los valores de la desautenticación han sido calculados con fórmula (4.14). Como se puede observar en las tablas C1-C6, los tiempos de desautenticación son tan pequeños respecto al tiempo de autenticación y los 60 s de espera, que los podemos considerar despreciables.

C.1. Latencias para el protocolo EAP-TLS.

Tabla C.1 Latencias de los diferentes escenarios para el protocolo EAP-TLS.

Retardo [s]	3Com Pc1	3Com Pc2	3Com Pc3	Intel Pc1	Atheros Pc2	Atheros Pc3	Cisco Pc1	Cisco Pc2
T.desautenticación	0,00003144	0,00003144	0,00003144	0,00003144	0,00003144	0,00003144	0,00003144	0,00003144
T.espera	60	60	60	60	60	60	60	60
T.autenticación	0,333976	0,222864	0,171206	0,291994	0,191367	0,289953	0,288383	0,199865
Latencia	60,33	60,23	60,17	60,29	60,19	60,29	60,29	60,2

C.2. Latencias para el protocolo EAP-PEAP

Tabla C2. Latencias de los diferentes escenarios para el protocolo EAP-PEAP.

Retardo [s]	3Com Pc1	3Com Pc2	3Com Pc3	Intel Pc1	Atheros Pc2	Atheros Pc3	Cisco Pc1	Cisco Pc2
T.desautenticación	0,00003144	0,00003144	0,00003144	0,00003144	0,00003144	0,00003144	0,00003144	0,00003144
T.espera	60	60	60	60	60	60	60	60
T.autenticación	0,817882	0,520944	0,53067	0,690996	0,579955	0,728339	0,619237	0,6159
Latencia	60,82	60,52	60,53	60,69	60,58	60,73	60,62	60,62

C.3. Latencias para el protocolo EAP-TTLS

Tabla C3. Latencias de los diferentes escenarios para el protocolo EAP-TTLS

Retardo [s]	3Com Pc1	3Com Pc2	3Com Pc3	Intel Pc1	Atheros Pc2	Atheros Pc3	Cisco Pc1	Cisco Pc2
T.desautenticación	0,00003248	0,00003248	0,00003248	0,00003248	0,00003248	0,00003248	0,00003248	0,00003248
T.espera	60	60	60	60	60	60	60	60
T.autenticación	0,286501	0,28271	0,177967	0,250262	0,176747	0,309078	0,273555	0,178699
Latenia	60,29	60,28	60,18	60,25	60,18	60,31	60,27	60,19

C.4. Latencias para el protocolo EAP-TTLS (software Intel)

La tarjeta Intel contiene un software que permite trabajar con los diferentes métodos de autenticación PAP, CHAP, MSCHAP y MSCHAPv2.

Tabla C.4 Latencias de los diferentes escenarios para el protocolo EAP-TTLS empleando el software de Intel.

Retardo [s]	PAP	CHAP	MSCHAP	MSCHAPv2
T.desautenticación	0,00003144	0,00003144	0,00003144	0,00003144
T.espera	60	60	60	60
T.autenticación	0,141844	0,136284	0,134425	0,164326
Latencia	60,14	60,14	60,13	60,16

C.5. Latencias para el protocolo EAP-LEAP

Tabla C.5 Latencias de los diferentes escenarios para el protocolo EAP-LEAP

Retardo [s]	Intel Pc1	Atheros Pc2	Cisco Pc1	Cisco Pc2
T.desautenticación	0,00003144	0,00003144	0,00003144	0,00003144
T.espera	60	60	60	60

T.autenticación	0,142305	0,0666	0,193171	0,070692
Latencia	60,14	60,1	60,19	60,07

C.6. Latencias del protocolo EAP-LEAP (AP + Servidor Radius)

En este apartado se mostrará las latencias de los diferentes escenarios con el protocolo EAP-LEAP teniendo como autenticador un servidor RADIUS.

Tabla C.6 Latencias de los diferentes escenarios para el protocolo EAP-LEAP con el servidor Radius

Retardo [s]	Intel Pc1	Atheros Pc2	Cisco Pc1	Cisco Pc2
T.desautenticación	0,00003144	0,00003144	0,00003144	0,00003144
T.espera	60	60	60	60
T.autenticación	0,08089	0,058477	0,151627	0,060526
Latencia	60,08	60,06	60,15	60,06

D. Código para las STAs atacadas

Para poder simular los escenarios con STAs atacadas se modificó la función *PaqNodo (*que genera los siguientes paquetes a transmitir*)* del fichero nodo.cpp

PaqNodo se modifica de forma que, a través de los identificadores de las STAs (idNodo), se le indica qué STAs serán atacadas impidiéndoles transmitir durante un tiempo de ataque. Para determinar el tiempo de ataque y el periodo de ataque, se empleará el uso de dos contadores (tnodo_contador1 y tnodo_contador 2).

```
int aux = su->IdNodo(); //  se guarda en la variable aux el id del nodo

long double i=0;


if(aux==1)||(aux==2)) // Se determina que la STA con id 1 y 2 serán las
   atacadas

  {


if(tnodo-tnodo_contador1>=periodo_ataque)  // (tnodo-tnodo_contador1)
   nos indica  cuando una estación entra en estado de atacada y se le  impide
   transmitir  durante un tiempo de ataque
{

 paqtu.tactual=cte.TSIMULACION+1; // Impide la transmisión.

      if (actualizar)
{
   tnodo_contador2=tnodo; // Se actualiza el valor de tnodo_contador2
   usándolo de referencia para saber cuándo termina el ataque.

    actualizar = FALSE;
}

if((tnodo-tnodo_contador2)>=tiempo_ataque) // la diferencia entre el tnodo y
   tnodo_contador2 nos indicará si la estación ha dejado de ser atacada (si ha
   transcurrido el tiempo de ataque), para así poder transmitir.

  {

  tnodo_contador1=tnodo - tiempo_ataque;// se actualiza tnodo_contador1
   actualizar = TRUE;
```

```
            paqtu.tactual=tnodo; // Ya puede transmitir

                //Se calcula el tiempo entre paquetes

             paqtu.tactual=tnodo + tproceso;
            paqtu.tcreacion=tnodo;
             paqtu.idtu=idnodo;          //Dentro de una LAN identifica al generador
             paqtu.origen=su->IdNodo();   //Identificador del nodo origen del mac
             paqtu.destino=au->IdNodo();  //El SU, función del AU que tenga asociado
             paqtu.age=INFINITO;
             paqtu.tipo=GEN;             //Para indicar que proviene de un TU
             paqtu.tpaq=0;               //Por defecto, se actualizará función del #bytes
                                  y la modulación

             paqtu.numheader=cte.HEADER_DATA; //Según el estándard

}

         }

       else // Si la STA atacada aun no está siendo atacada puede transmitir

  {
            //Se calcula el tiempo entre paquetes

             paqtu.tactual=tnodo + tproceso;
             paqtu.tcreacion=tnodo;
             paqtu.idtu=idnodo;          //Dentro de una LAN identifica al generador
             paqtu.origen=su->IdNodo();   //Identificador del nodo origen del mac
             paqtu.destino=au->IdNodo();  //El SU, función del AU que tenga asociado
             paqtu.age=INFINITO;
             paqtu.tipo=GEN;             //Para indicar que proviene de un TU
             paqtu.tpaq=0;               //Por defecto, se actualizará función del #bytes y la
modulación
             paqtu.numheader=cte.HEADER_DATA; //Según el estándard


            }
        }
     else // Si es una STA no atacada puede transmitir
         {
            //Se calcula el tiempo entre paquetes

             paqtu.tactual=tnodo + tproceso;
             paqtu.tcreacion=tnodo;
             paqtu.idtu=idnodo;          //Dentro de una LAN identifica al generador
             paqtu.origen=su->IdNodo();   //Identificador del nodo origen del mac
             paqtu.destino=au->IdNodo();  //El SU, función del AU que tenga asociado
             paqtu.age=INFINITO;
             paqtu.tipo=GEN;             //Para indicar que proviene de un TU
```

```
            paqtu.tpaq=0;              //Por defecto, se actualizará función del #bytes
                                         y la modulación
            paqtu.numheader=cte.HEADER_DATA; //Según el estándard

           }



      idpaq++;
      paqtu.idpaq=idpaq;

    return p;


     }
```

E. Código modificado modelo analítico.

Para realizar los cálculos de los dos modelos analíticos se añadieron dos funciones en parámetros. Cpp del proyecto modelo, la función calculo_tau y la función de Calculo_P. Estas funciones son necesarias para calcular los parámetros P_{tr} y P_s definidos en el modelo analítico de Bianchi [1]. Una vez obtenidos los valores de calculo_tau y calculo_P, mediante una hoja de cálculo de Excel, ya se pueden calcular los valores del throughput (*S).*

- **Modelo. Cpp**

```
//---------------------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop

USERES("modelo.res");
USEUNIT("parametros.cpp");

#include <stdlib.h>

#include "parametros.h"
#include "Ctes.h"

//---------------------------------------------------------------------------


WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
{
   long double p1[num_estaciones];

   long double tau[num_estaciones];
   long double p2[num_estaciones];
   long double pi[num_estaciones];
   long double pj[num_estaciones];

   bool conv[num_estaciones];

   PARAMETROS* parametros = new PARAMETROS;

   for (int i=0; i<num_estaciones; i++)
   {
      p1[i]=0.0;
      tau[i]=0.0;
      p2[i]=0.0;
      pi[i]=0.0;
      pj[i]=0.0;
   }
```

```
    for (int i=100; i<=iteraciones; i++)
    {
       p1[0]=i*inc;
       p1[1]=i*inc;
                                  for (int v=0; v<num_estaciones; v++)
                                  {
                                     tau[v]=parametros->CalculoTau(p1[v]);

                                  }

                                  for (int v=0; v<num_estaciones; v++)
                                     p2[v]=parametros->CalculoP(tau[v]);


                                  for (int v=0; v<num_estaciones; v++)
                                     conv[v]=false;

                                  for (int v=0; v<num_estaciones; v++)
                                  {
                                     long double aux = p1[v]-p2[v];

                                     if (aux < 0) aux *=-1;

                                     if (aux > error)
                                        break; // del for v
                                     else
                                        conv[v]=true;
                                  }

                                   if (conv[0] && conv[1])

                                     {
                                          bool stop = true;
                                       }

                                }

    return 0;
}
```

- **Parámetros. Cpp**

```
//---------------------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop

#include "parametros.h"

//---------------------------------------------------------------------------
#pragma package(smart_init)

#include    <stdlib.h>
#include    <stdio.h>
#include    <dir.h>
#include    <math.h>

#include    "Ctes.h"

// Constructor de la clase PARAMETROS

PARAMETROS::PARAMETROS(void)
{
}

long double PARAMETROS::CalculoTau (long double p)
{
   /*long double suma=0;

   for(int i=0;i<m;i++)
      suma=suma+powl(2*p,i);

   suma=suma+powl(2.0*p,m)/(1.0-p);
   suma=W*suma;
   suma=suma+1/(1-p);
   suma=0.5*suma;
   suma=(1-p)*suma;

   return 1.0/suma; */
    long double a=0;
    long double b=0;
    long double b1=0;
    long double b2=0;


    a=2*(1-2*p);
    b1=(1-2*p)*(W+1);
    b2=p*W*(1-powl(2.0*p,m));
    b=b1+b2;
```

```
        return a/b;


}

/*


*/

long double PARAMETROS::CalculoP (long double tau)
{

    long double p=1;


     p=1-(powl((1-tau),num_estaciones-1));

    return p;


}
```

F. Resultados del modelo analítico

- **Periodo de ataque 181,002s.**

Tabla F.1. Resultados para el periodo 181,002s.

Número de estaciones	Tamaño de la trama	Steórico	Spráctico	% diferencia
2	200	0,105	0,122	13,347
4	200	0,111	0,122	8,959
10	200	0,109	0,117	7,079
15	200	0,106	0,114	7,015
20	200	0,104	0,111	6,702
50	200	0,095	0,102	6,799
100	200	0,086	0,093	7,082
2	600	0,237	0,269	12,009
4	600	0,240	0,263	8,535
10	600	0,229	0,247	7,348
15	600	0,221	0,239	7,487
20	600	0,216	0,233	7,258
50	600	0,194	0,210	7,537
100	600	0,174	0,189	7,882
2	1000	0,319	0,355	10,008
4	1000	0,318	0,342	6,972
10	1000	0,298	0,318	6,133
15	1000	0,287	0,307	6,336
20	1000	0,279	0,297	6,089
50	1000	0,249	0,266	6,436
100	1000	0,223	0,239	6,758
2	1500	0,380	0,425	10,664
4	1500	0,373	0,406	8,192
10	1500	0,345	0,374	7,703
15	1500	0,331	0,360	8,032
20	1500	0,321	0,348	7,799
50	1500	0,285	0,310	8,246
100	1500	0,253	0,277	8,608

- **Periodo de ataque 301,67 s**

Tabla F2. Resultados para el periodo 301,67 s

Número de	Tamaño de la	Steórico	Spráctico	%

estaciones	trama			diferencia
2	200	0,132	0,145	8,967
4	200	0,138	0,145	5,318
10	200	0,134	0,139	4,024
15	200	0,130	0,136	4,087
20	200	0,128	0,133	3,882
50	200	0,117	0,122	4,246
100	200	0,106	0,111	4,671
2	600	0,293	0,321	8,652
4	600	0,295	0,313	5,817
10	600	0,279	0,294	5,101
15	600	0,270	0,285	5,338
20	600	0,263	0,277	5,186
50	600	0,236	0,250	5,694
100	600	0,212	0,226	6,091
2	1000	0,392	0,423	7,183
4	1000	0,388	0,408	4,670
10	1000	0,363	0,379	4,198
15	1000	0,349	0,351	0,545
20	1000	0,339	0,354	4,327
50	1000	0,302	0,317	4,817
100	1000	0,270	0,285	5,273
2	1500	0,465	0,507	8,333
4	1500	0,453	0,484	6,316
10	1500	0,419	0,446	6,120
15	1500	0,401	0,429	6,534
20	1500	0,389	0,415	6,328
50	1500	0,344	0,370	6,883
100	1500	0,306	0,330	7,343

- **Periodo de ataque 603,34 s**

Tabla F3. Resultados para el periodo 603,34 s

Número de estaciones	Tamaño de la trama	Steórico	Spráctico	% diferencia
2	200	0,152	0,163	6,498
4	200	0,158	0,163	3,267
10	200	0,153	0,156	2,302
15	200	0,149	0,152	2,457
20	200	0,145	0,149	2,309
50	200	0,133	0,136	2,813
100	200	0,120	0,124	3,301
2	600	0,335	0,359	6,754

4	600	0,336	0,351	4,291
10	600	0,317	0,330	3,818
15	600	0,306	0,320	4,155
20	600	0,298	0,311	3,999
50	600	0,268	0,281	4,580
100	600	0,240	0,253	5,120
2	1000	0,447	0,474	5,607
4	1000	0,441	0,457	3,401
10	1000	0,411	0,424	3,118
15	1000	0,395	0,410	3,517
20	1000	0,384	0,397	3,412
50	1000	0,342	0,356	3,947
100	1000	0,305	0,319	4,404
2	1500	0,528	0,568	7,030
4	1500	0,514	0,542	5,271
10	1500	0,474	0,499	5,192
15	1500	0,454	0,481	5,685
20	1500	0,440	0,465	5,521
50	1500	0,389	0,415	6,146
100	1500	0,346	0,370	6,617

- **Periodo de ataque 1206,68 s**

Tabla F4. Resultados para el periodo 1206,68 s

Número de estaciones	Tamaño de la trama	Steórico	Spráctico	% diferencia
2	200	0,162	0,168	3,812
4	200	0,168	0,169	0,722
10	200	0,162	0,162	0,126
15	200	0,158	0,158	0,071
20	200	0,154	0,154	0,060
50	200	0,141	0,141	0,468
100	200	0,127	0,129	1,036
2	600	0,356	0,372	4,345
4	600	0,357	0,364	1,959
10	600	0,336	0,342	1,618
15	600	0,325	0,331	1,989
20	600	0,316	0,322	1,829
50	600	0,283	0,291	2,487
100	600	0,254	0,262	3,036
2	1000	0,475	0,491	3,276
4	1000	0,468	0,473	1,171
10	1000	0,435	0,440	1,010

15	1000	0,418	0,424	1,393
20	1000	0,406	0,411	1,268
50	1000	0,362	0,369	1,883
100	1000	0,323	0,330	2,370
2	1500	0,560	0,589	4,869
4	1500	0,544	0,562	3,168
10	1500	0,501	0,518	3,210
15	1500	0,480	0,498	3,680
20	1500	0,465	0,482	3,513
50	1500	0,412	0,430	4,195
100	1500	0,365	0,383	4,671

- **Periodo de ataque 1800 s**

Tabla F5. Resultados para el periodo 1800 s

Número de estaciones	Tamaño de la trama	Steórico	Spráctico	% diferencia
2	200	0,165	0,174	5,148
4	200	0,171	0,175	2,156
10	200	0,165	0,168	1,365
15	200	0,161	0,163	1,577
20	200	0,157	0,160	1,463
50	200	0,143	0,146	2,013
100	200	0,130	0,133	2,570
2	600	0,363	0,385	5,738
4	600	0,363	0,376	3,446
10	600	0,343	0,354	3,135
15	600	0,331	0,343	3,524
20	600	0,322	0,333	3,360
50	600	0,289	0,301	4,047
100	600	0,259	0,271	4,547
2	1000	0,484	0,508	4,731
4	1000	0,476	0,490	2,713
10	1000	0,443	0,455	2,538
15	1000	0,426	0,439	2,962
20	1000	0,414	0,425	2,793
50	1000	0,368	0,381	3,445
100	1000	0,328	0,342	3,935
2	1500	0,571	0,609	6,334
4	1500	0,554	0,581	4,709
10	1500	0,510	0,536	4,767
15	1500	0,489	0,516	5,231
20	1500	0,473	0,498	5,048

50	1500	0,419	0,444	5,742
100	1500	0,372	0,397	6,238

G. Influencia del número de STAs y el periodo del ataque para el escenario 3Com en pc1

- **3Com en pc1 con EAP-PEAP**

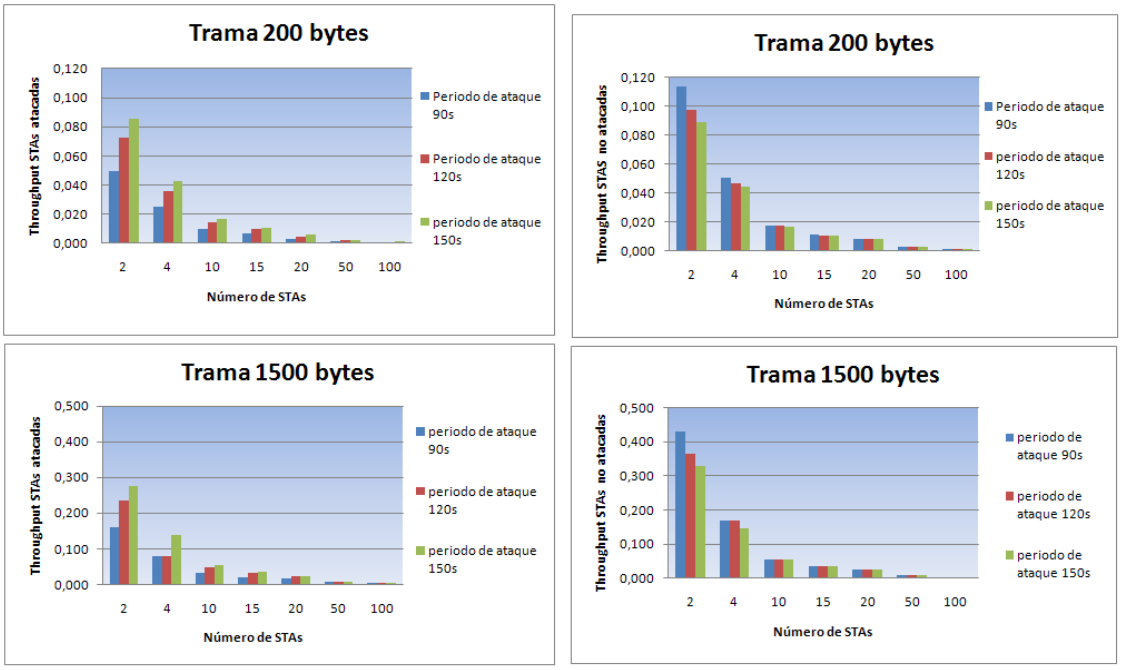


Fig. G1 3Com en pc1 con EAP-PEAP

- **3Com en pc1 con EAP-TTLS**

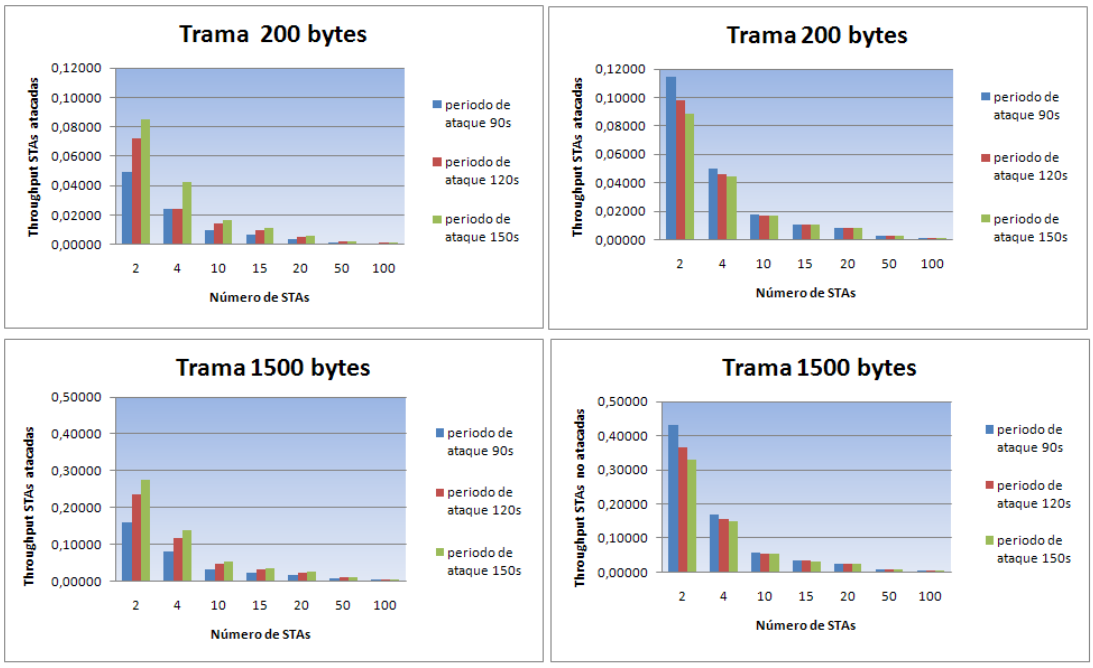


Fig. G2. 3Com en pc1 con EAP-TTLS

- **3Com en pc1 con EAP-TTLS (PAP)**

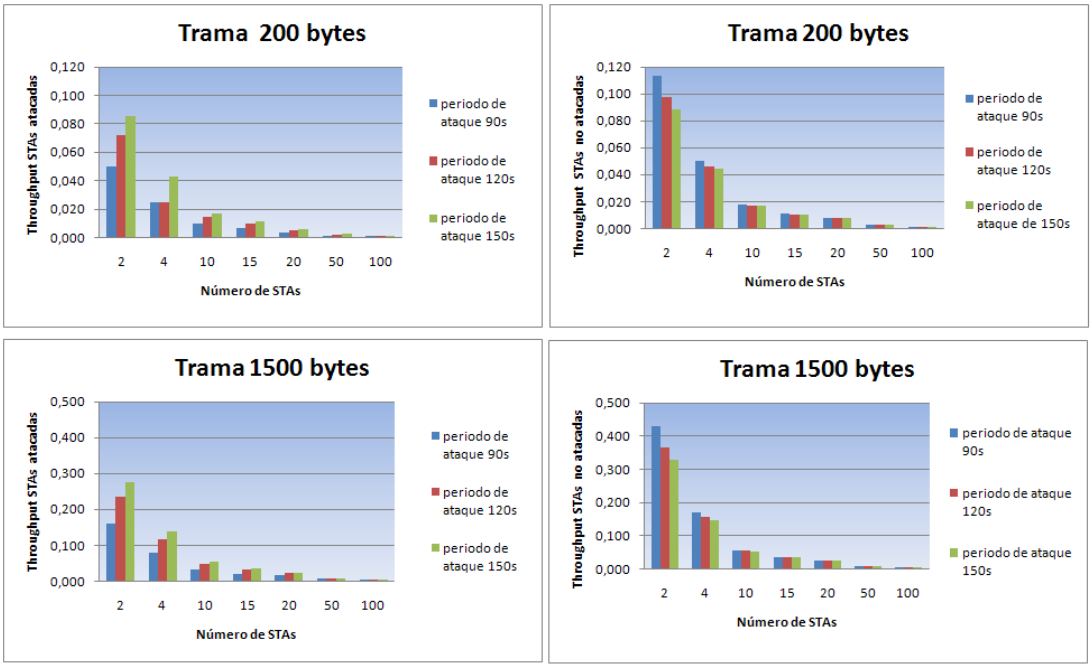


Fig. G3 3Com en pc1 utilizando método PAP

- **3Com en pc1 con EAP-TTLS (CHAP)**

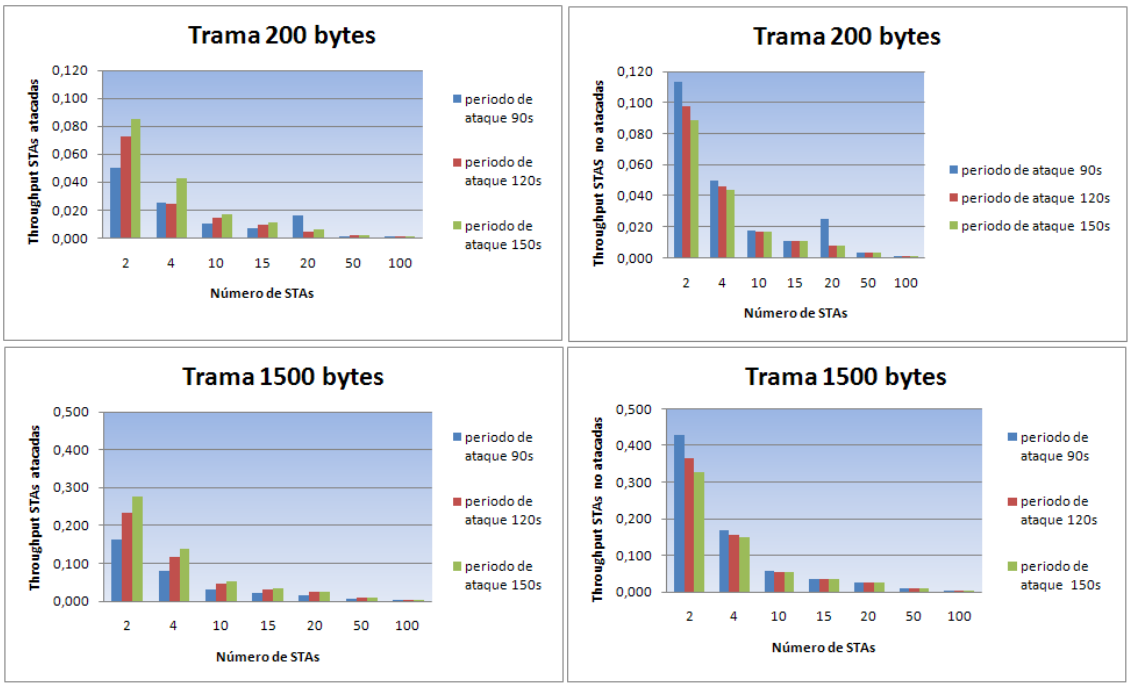


Fig. G4. 3Com en pc1 utilizando método CHAP

- **3Com en pc1 con EAP-TTLS (MSCHAP)**

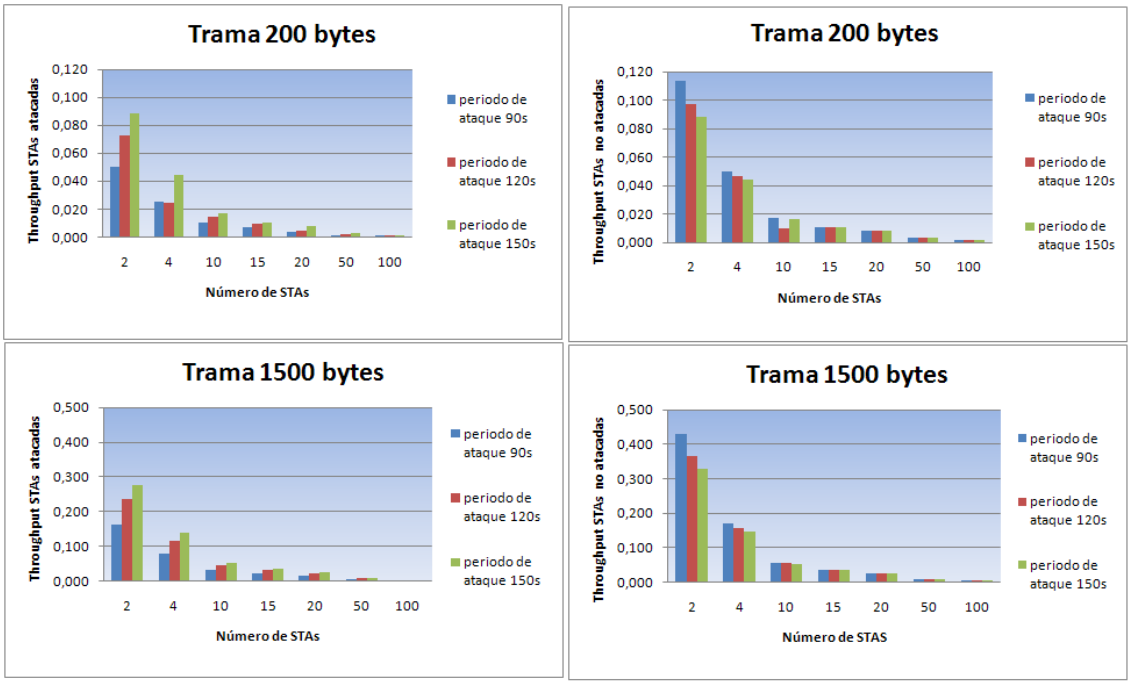


Fig. G5 3Com en pc1 utilizando método MSCHAP.

- **3Com en pc1 con EAP-TTLS (MSCHAPv2)**

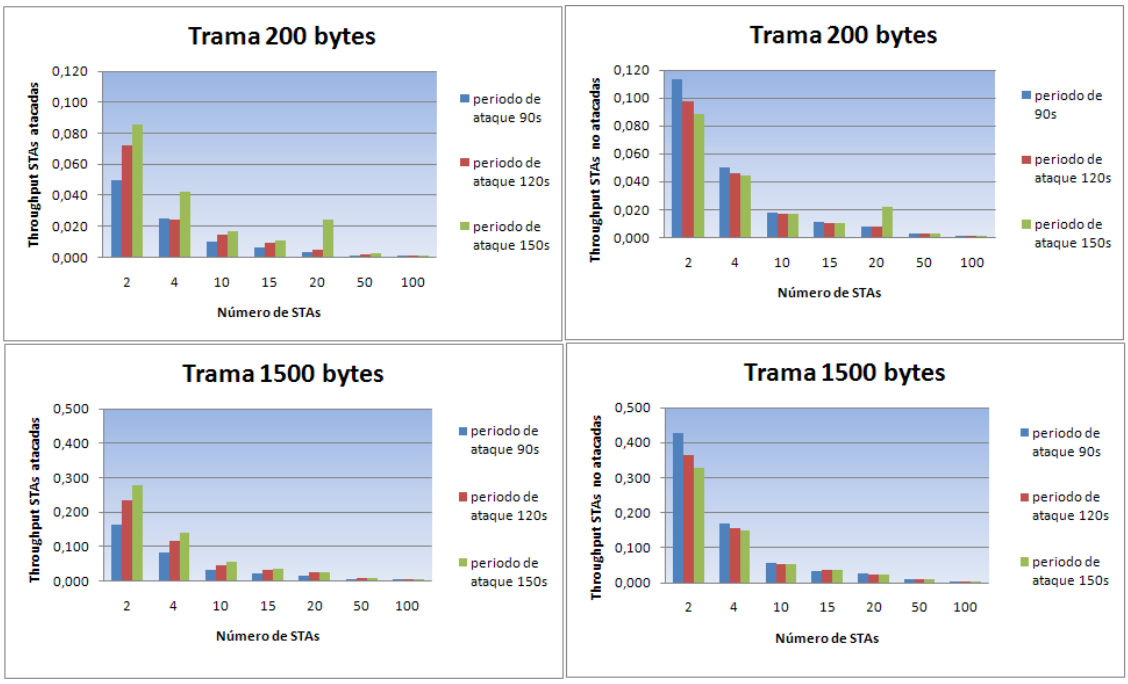


Fig.G6 3Com en pc1 utilizando método MSCHAPv2

- **3Com en pc1 con EAP-LEAP**

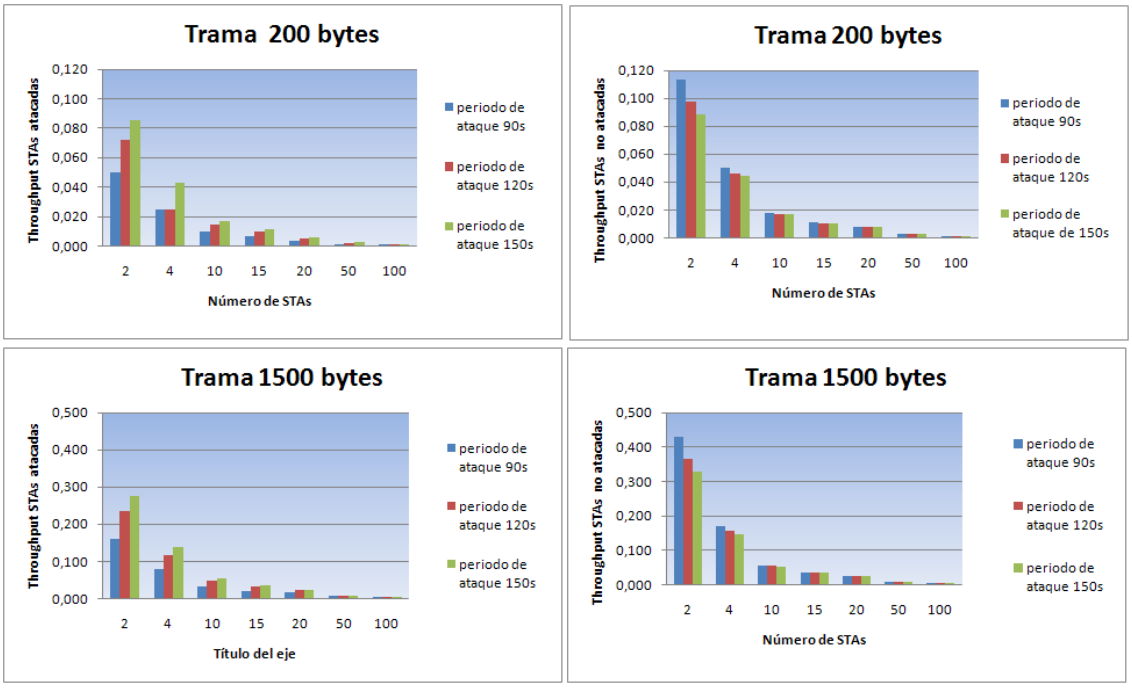


Fig. G7 3Com en pc1 con EAP-LEAP

- **3Com en pc1 con EAP-LEAP (AP+ servidor Radius)**

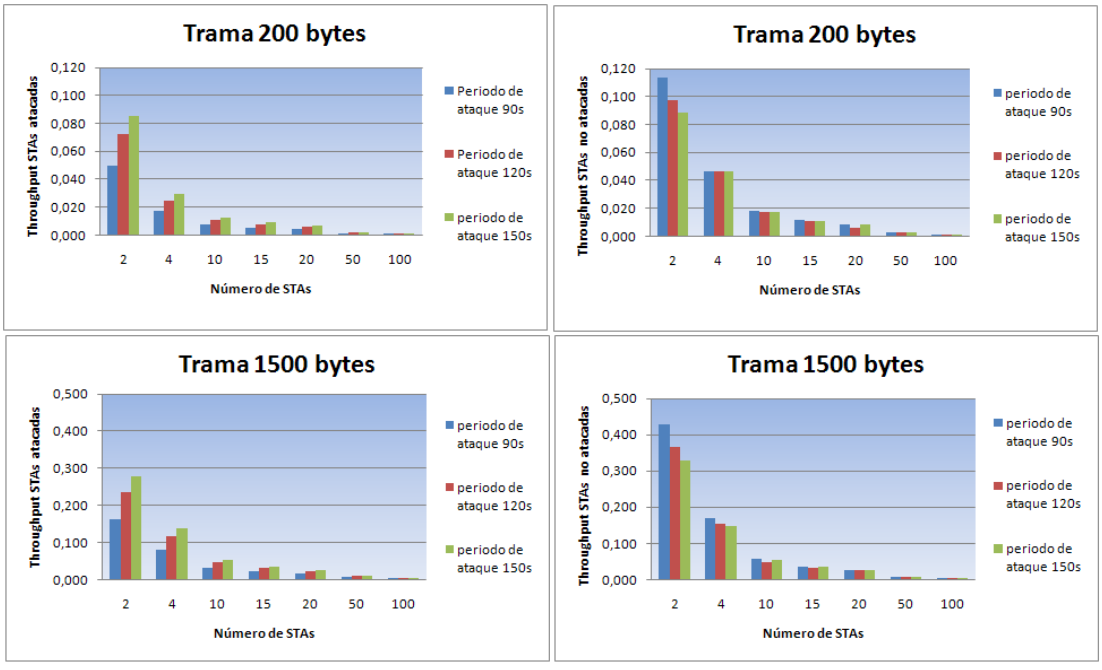


Fig. G8. 3Com en pc1 con EAP-LEAP(AP+servidor Radius)

H. Estudio de la justicia para el protocolo EAP-TLS

- **Tarjeta 3Com en pc1**

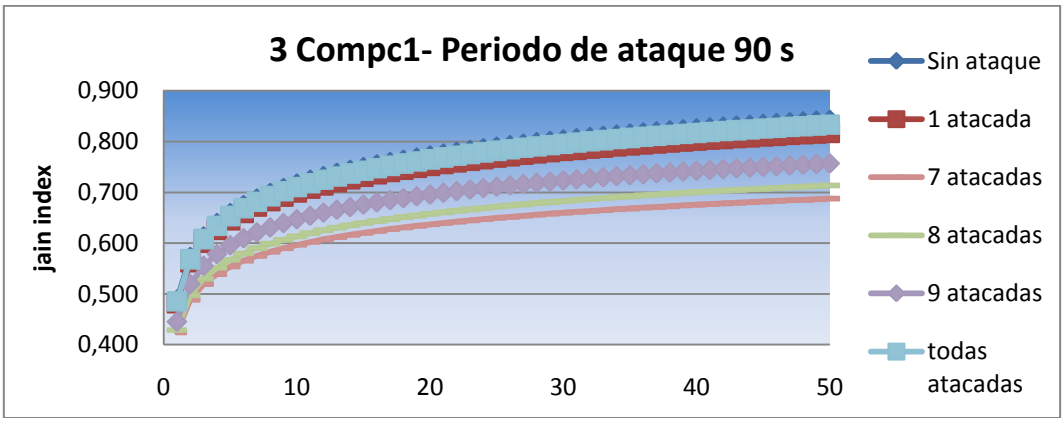


Fig. H1. Escenario 3Com en pc1 periodo de ataque 90s.

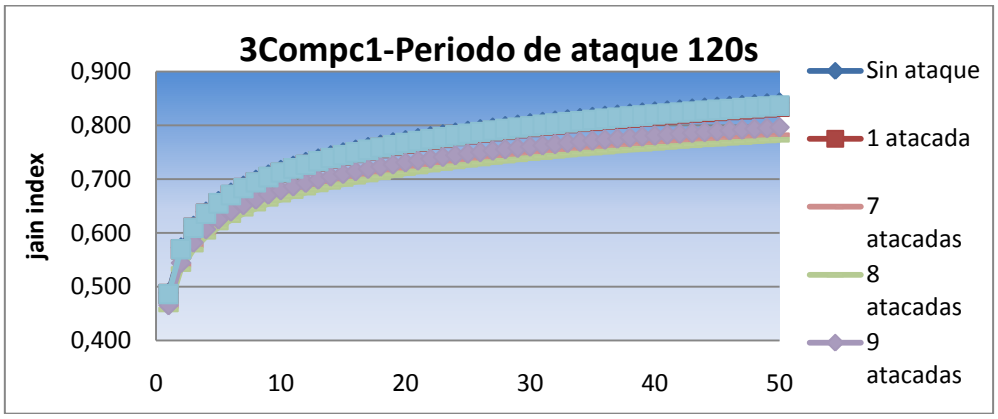


Fig. H2 Escenario 3Com en pc1 periodo de ataque de 120s.

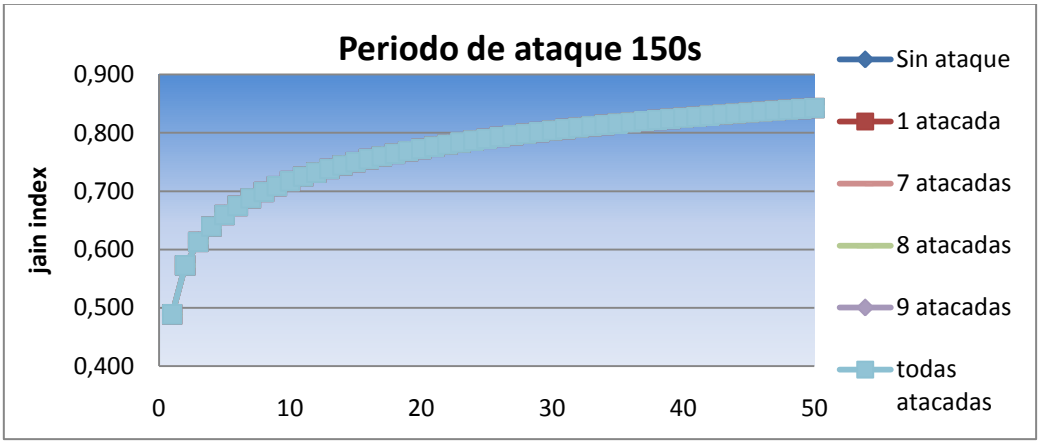


Fig. H3. Escenario 3Com en pc1 periodo de ataque 150s.

- **Tarjeta 3Com en pc2**

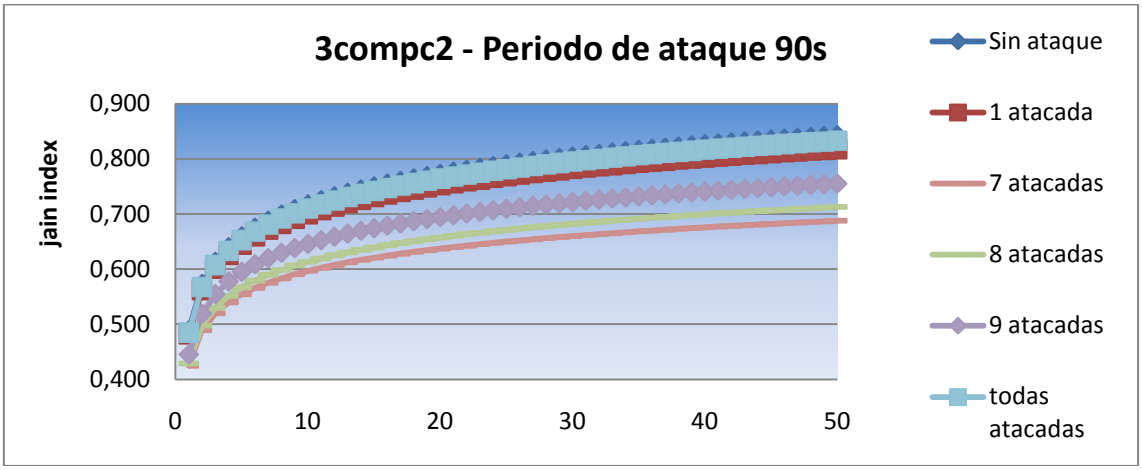


Fig. H4 Escenario 3Com en pc2 periodo de ataque 90s.

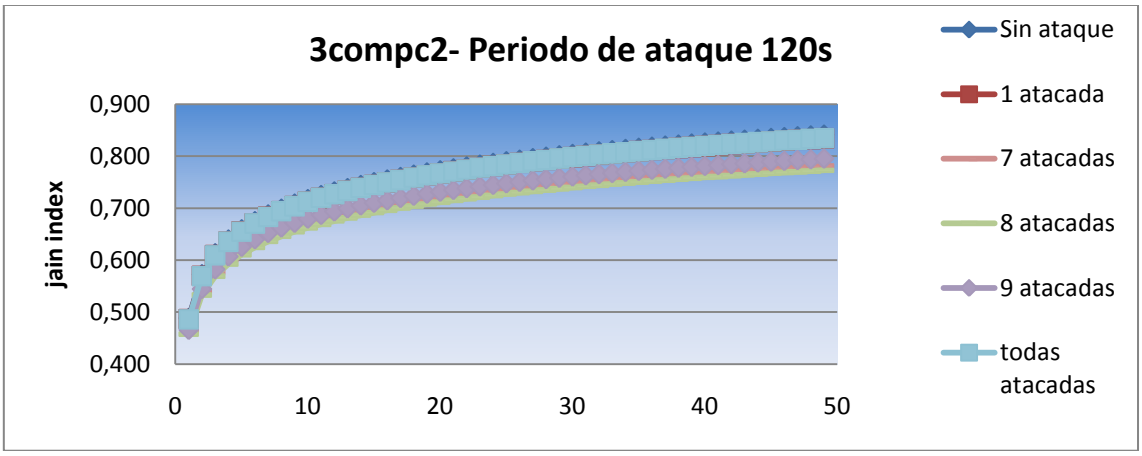


Fig. H5 Escenario 3Com en pc2 periodo de ataque 120s

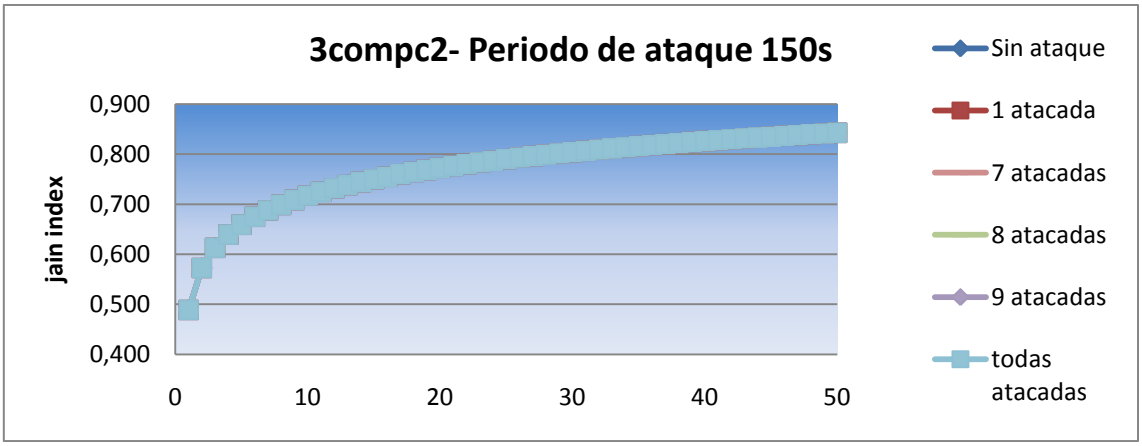


Fig. H6 Escenario 3Com en pc2 periodo de ataque 150s

- **Tarjeta 3Com en pc3**

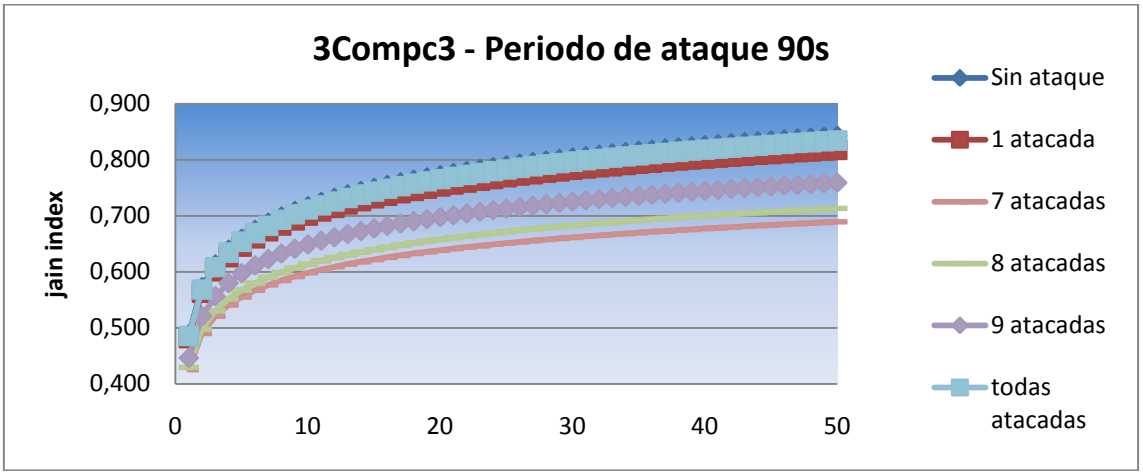


Fig. H7 Escenario 3Com en pc3 periodo de ataque 90s

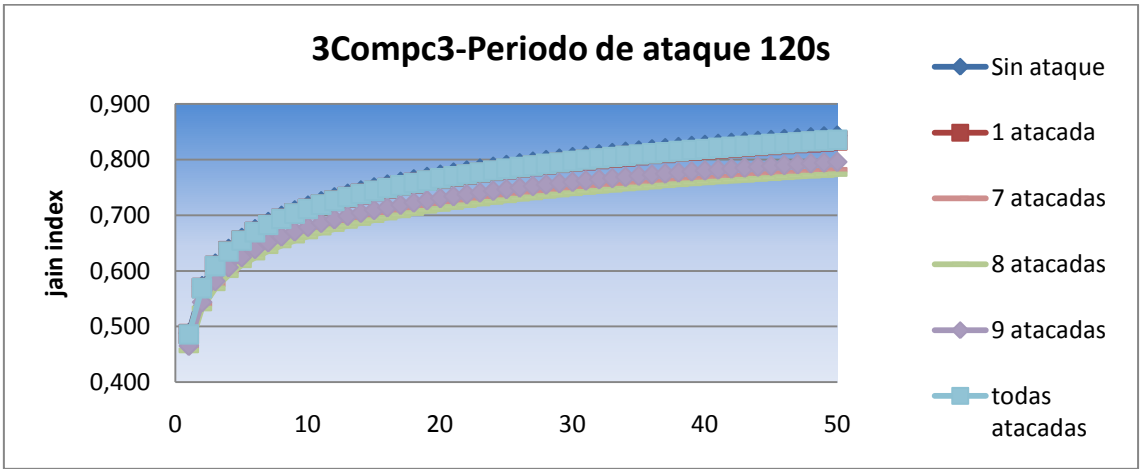


Fig. H8 Escenario 3Com en pc3 periodo de ataque 120s

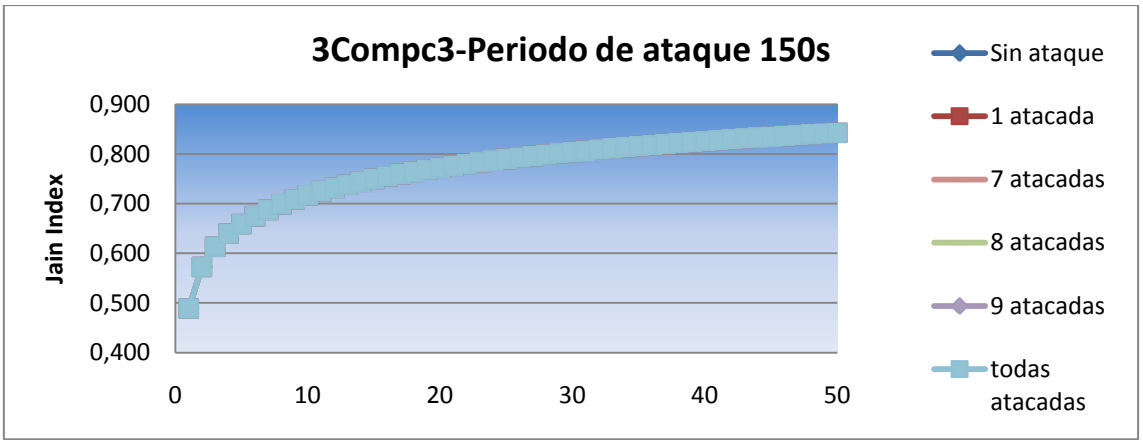


Fig. H9 Escenario 3Com en pc3 periodo de ataque 150s

- **Tarjeta Atheros en pc2**

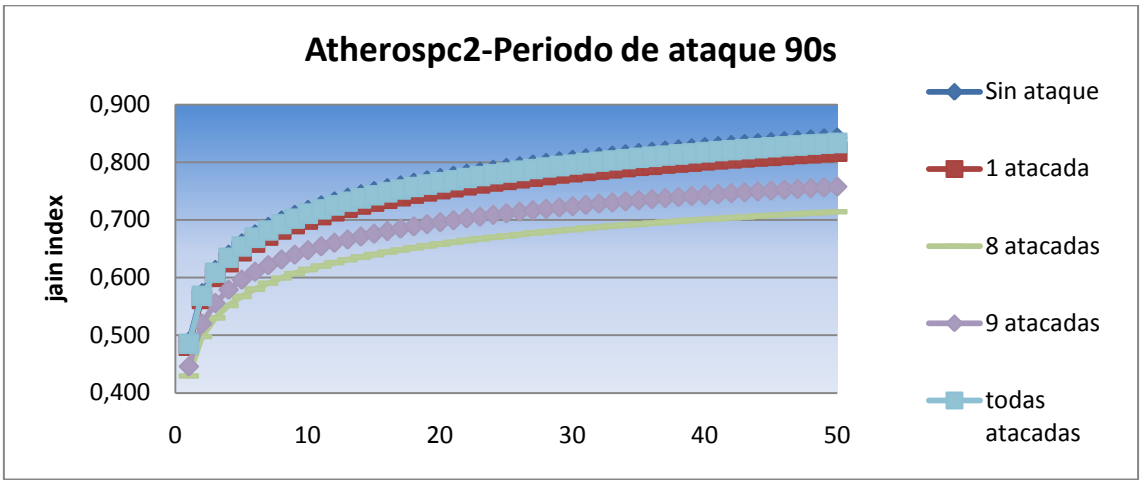


Fig H.10 Escenario Atheros en pc2 periodo de ataque 90s.

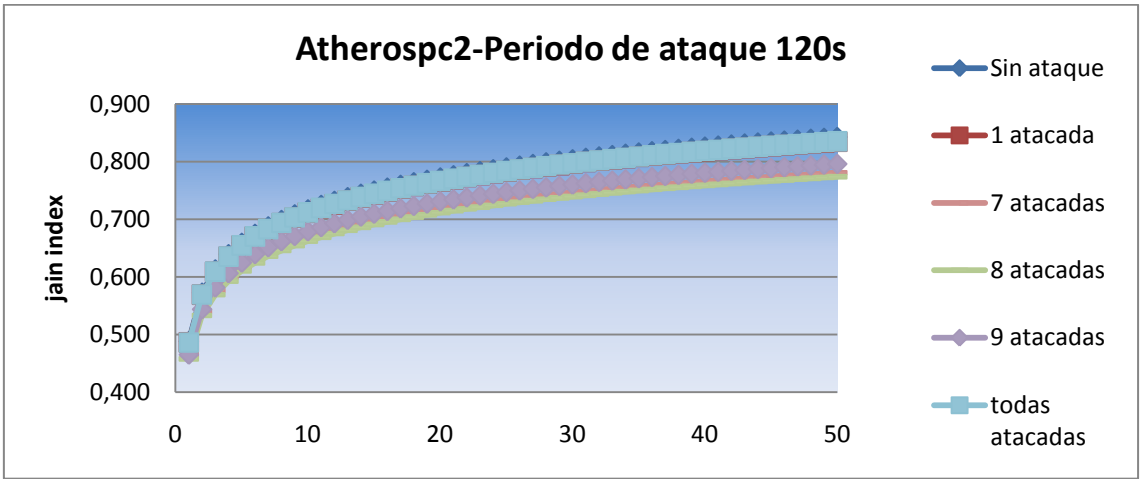


Fig H11. Escenario Atheros en pc2 periodo de ataque120s

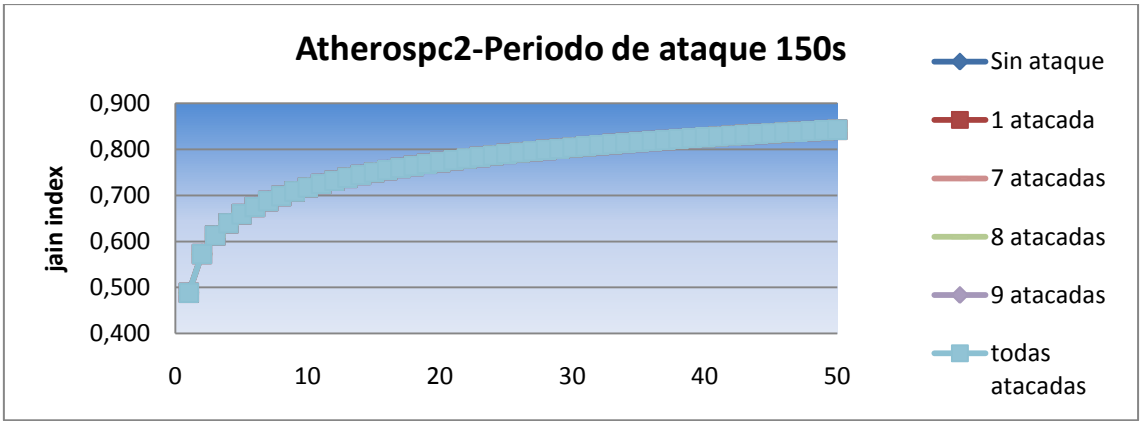


Fig. H12 Escenario Atheros en pc2 periodo de ataque 150s

- **Tarjeta Atheros en pc3**

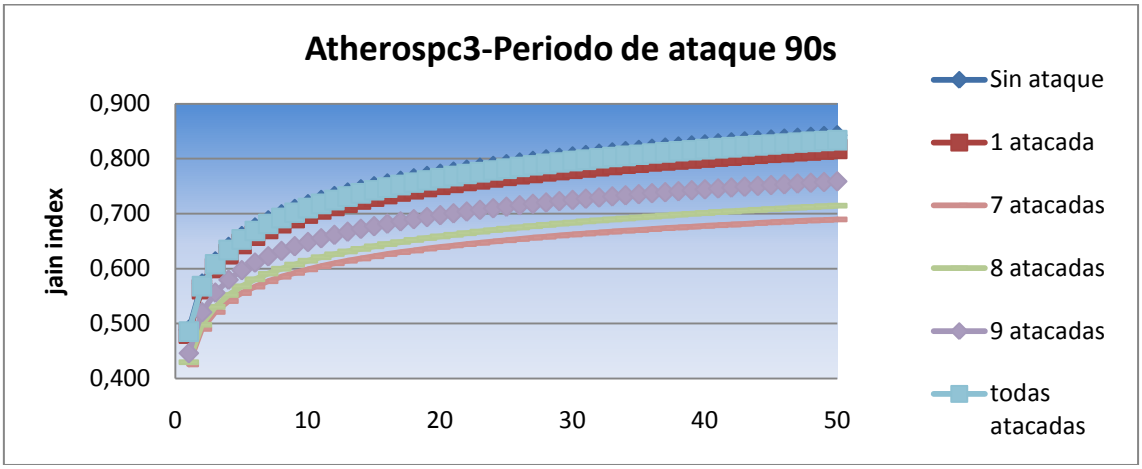


Fig. H13 Escenario Atheros en pc3 periodo de ataque 90s

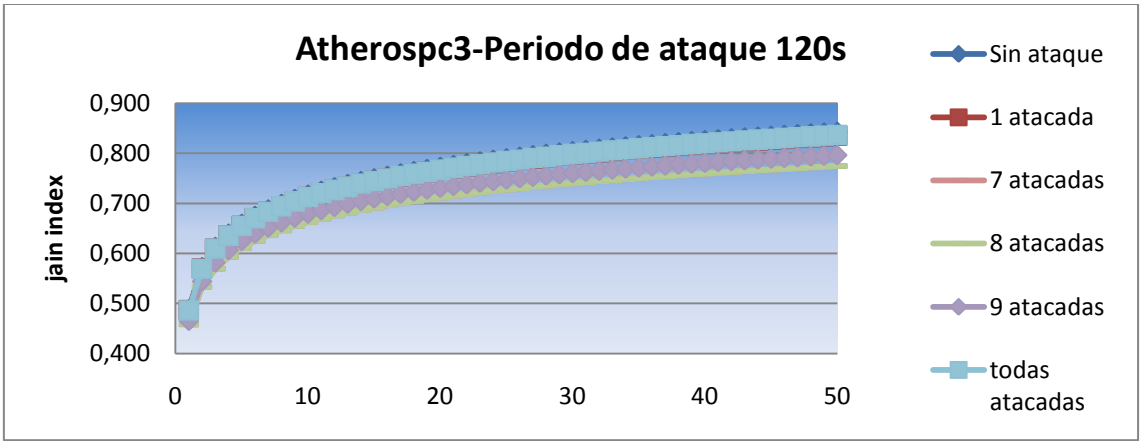


Fig. H14 Escenario Atheros en pc3 periodo de ataque 120s

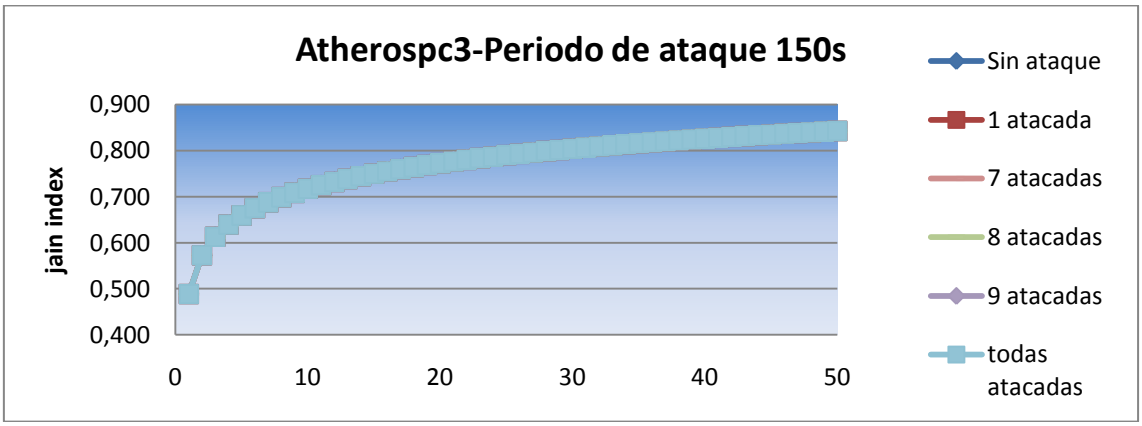


Fig. H15 Escenario Atheros en pc3 periodo de ataque 150s

- **Tarjeta Cisco en pc1**

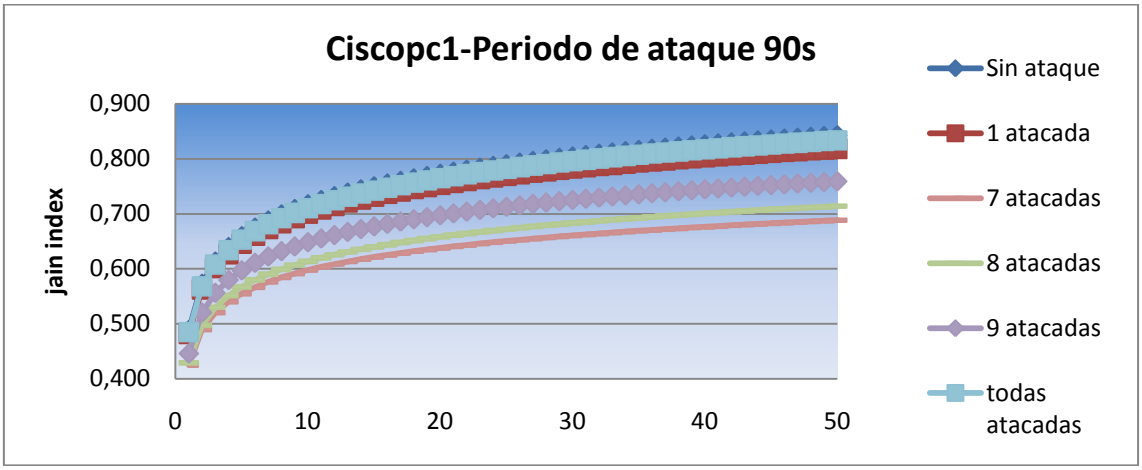


Fig. H16 Escenario Cisco en pc1 periodo de ataque 90s

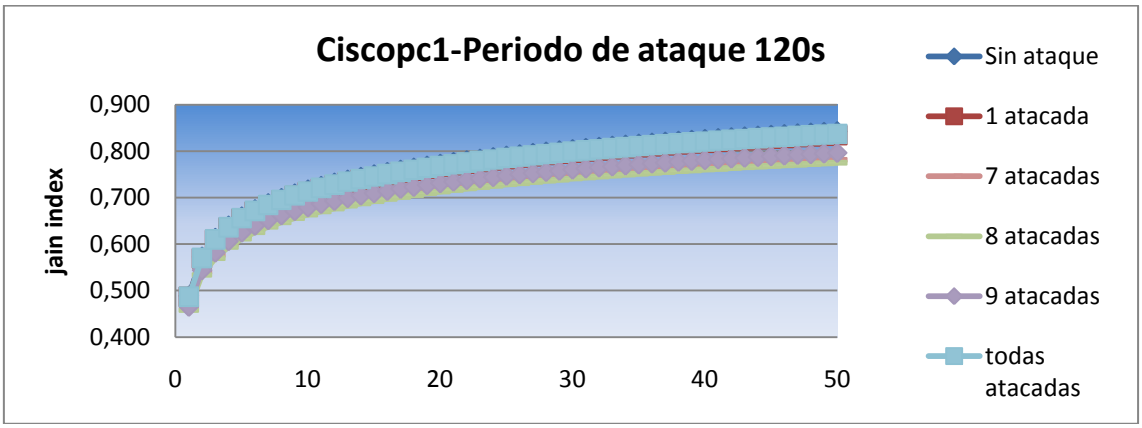


Fig. H17 Escenario Cisco en pc1 periodo de ataque 120s

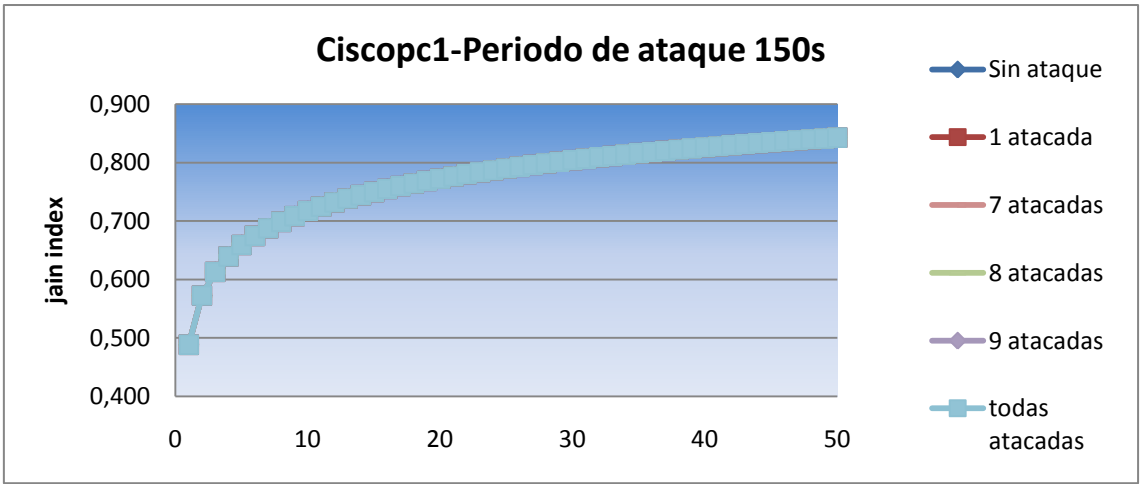


Fig. H18 Escenario Cisco en pc1 periodo de ataque 150s

- **Tarjeta Cisco en pc2**

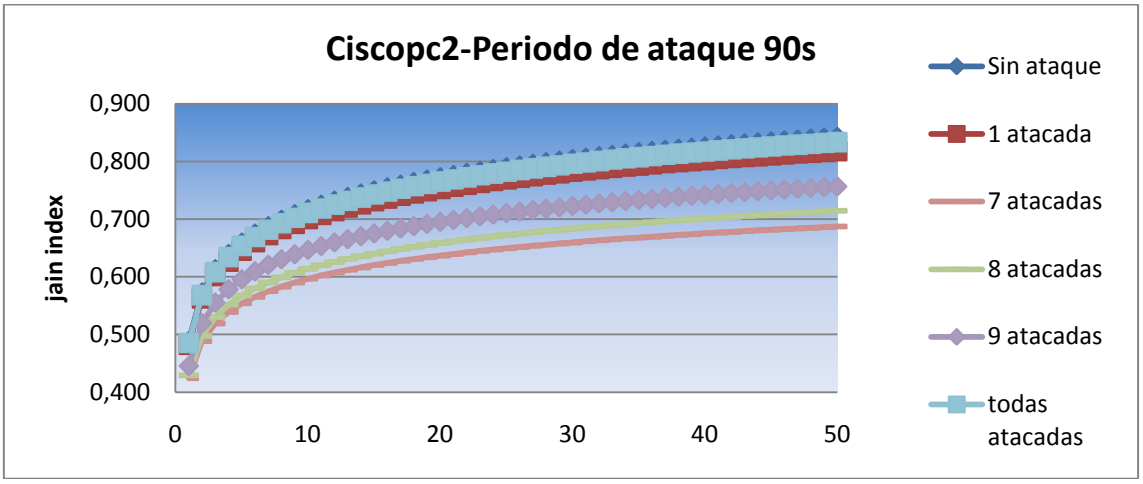


Fig.H19. Escenario Cisco en pc2 periodo de ataque 90s

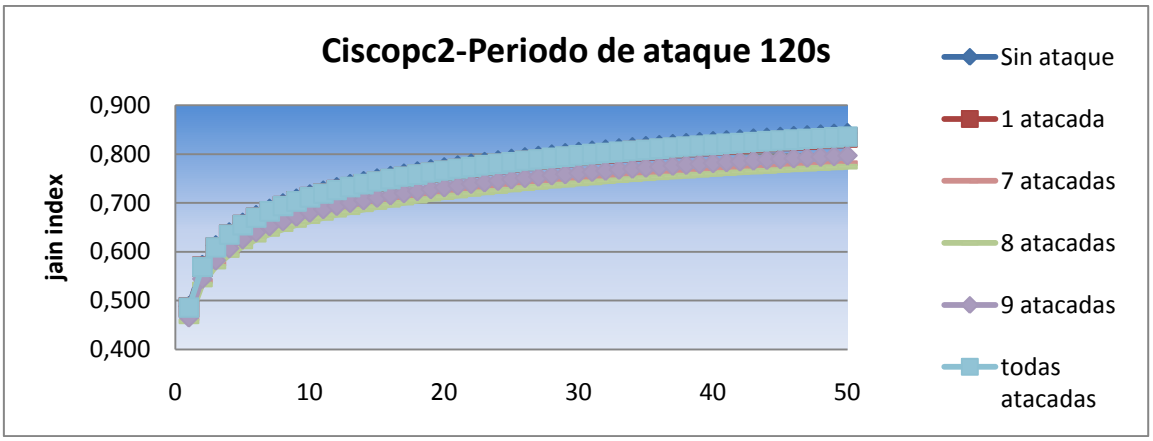


Fig.H20 Escenario Cisco en pc2 periodo de ataque 120s

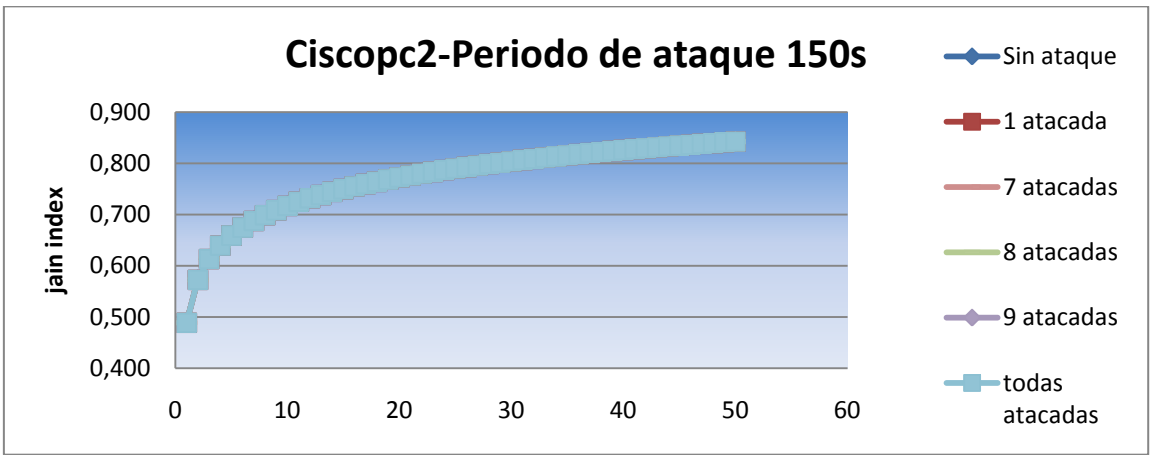


Fig. H21 Escenario Cisco en pc2 periodo de ataque 150s

I. Tiempo de backoff

Para el cálculo del tiempo de backoff se añadieron dos funciones al proyecto modelo, la función TBO y la función Ps. Para ello, se modificaron el fichero modelo.cpp y parámetros.cpp

- **Modelo.cpp**

```
for (int v=0; v<num_estaciones; v++)
 {
        pi[v]=(1-parametros->CalculoPs(tau[v]));
        pj[v]=parametros->TBO(pi[v]);
   }
```

- **Parametros.cpp**

```
long double PARAMETROS::CalculoPs (long double tau)
{
   long double x=0;
   long double y=0;

   x= num_estaciones*tau*(powl((1-tau),num_estaciones-1));
   y= 1-(powl((1-tau),num_estaciones));

   return x/y;

}

long double PARAMETROS::Tbo(long int i)
{
   long double ret;



   ret=((pow(2.0,i)*(CWmin+1.0)-1)/2.0);
   if(ret>CWmax/2.0)
      ret=(CWmax)/2.0;
   return ret*tSLOT;
   }

long double PARAMETROS::TBO(double prob)
{
```

```
        long int i;


    double prec, tmp, ret;

        prec=0.000001 ;// 1 microsegon
        i=0;
        tmp=10000.0;
        ret=0;
        while(tmp>prec)
        {
            tmp=(1.0-prob)*pow(prob,i);
            tmp=tmp*Tbo(i);
            ret+=tmp;
            i++;
        }
        return ret;
    }
```

J. Estudio del consumo de baterías

J.1. Consumo de las STAs atacadas

El consumo de las STAs atacadas viene predeterminado por el consumo del tiempo de ataque, comprendido entre los 60s de espera antes de la autenticación y la misma, descartando el tiempo de desautenticación por ser despreciable frente a los otros dos valores.

- **Consumo Intel en pc1 con el protocolo EAP-TLS**

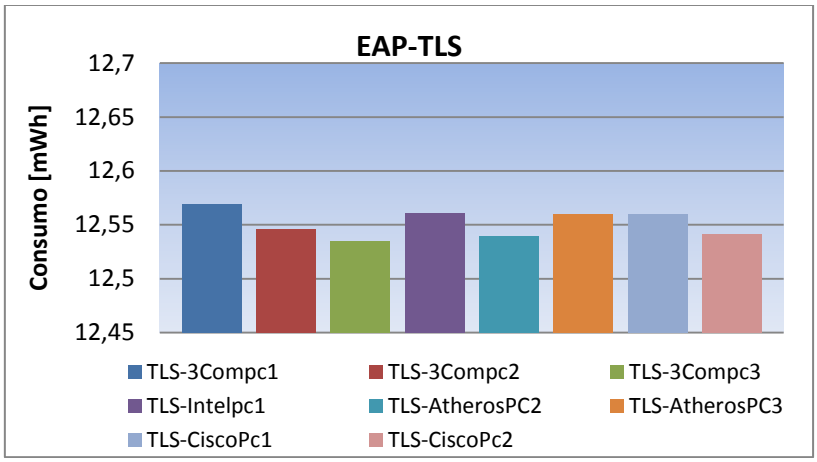


Fig.J1 Consumo de las STAs atacadas para EAP-TLS

- **Consumo Intel en pc1 con el protocolo EAP-PEAP**

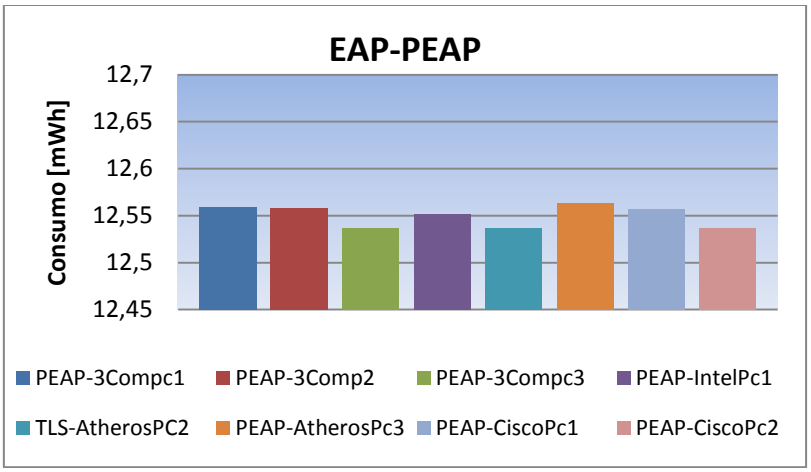


Fig. J2 Consumo de las STAs atacadas para EAP-PEAP

- **Consumo Intel en pc1 con el protocolo EAP-TTLS**

Se incluyen los valores de los diferentes métodos de autenticación que nos permite la tarjeta Intel en pc1.

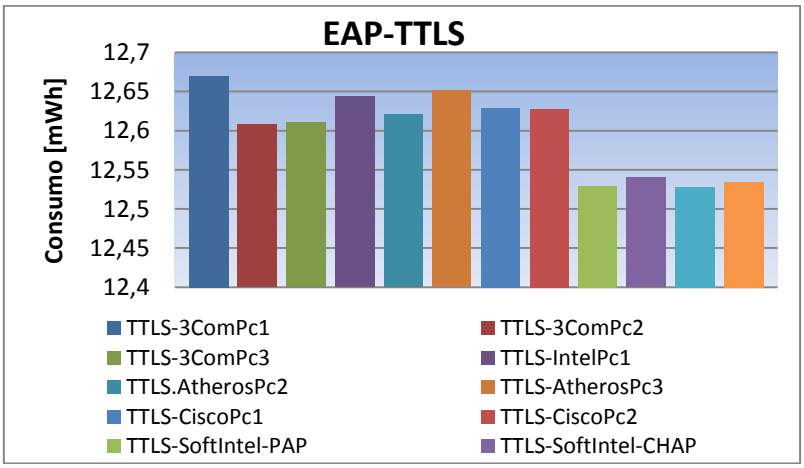


Fig. J3 Consumo de las STAs atacadas para EAP-TTLS

- **Consumo Intel en pc1 con el protocolo EAP-LEAP**

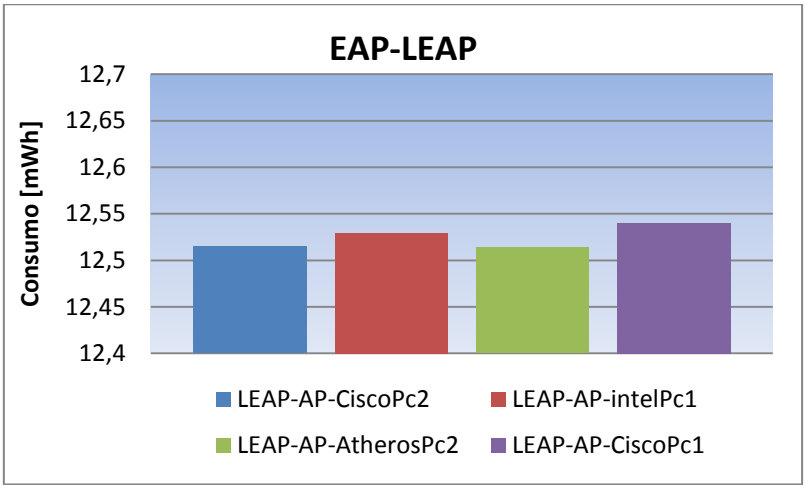


Fig. J4 Consumo de STAs atacadas para EAP-LEAP

- **Consumo Intel en pc1 con el protocolo EAP-LEAP (AP+servidor Radius)**

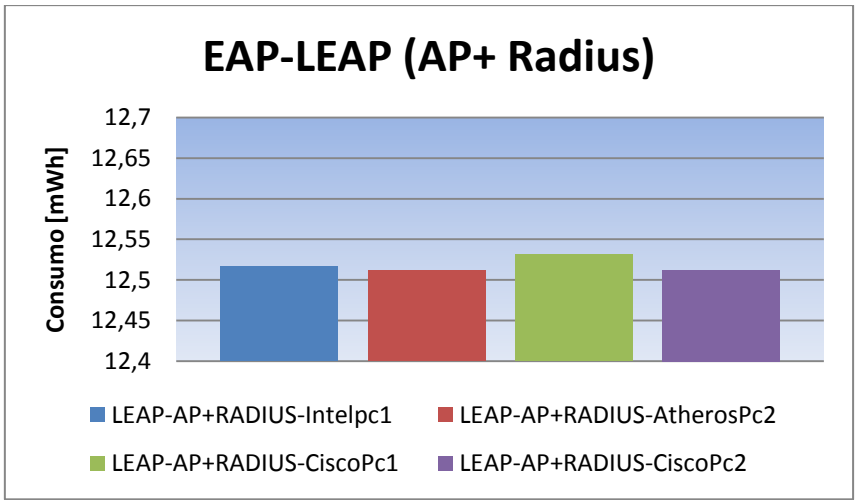


Fig J5. Consumo de las STAs atacadas para EAP-LEAP (AP+Servidor Radius)

- **Consumo de las STAs atacadas según el tipo de tarjeta**

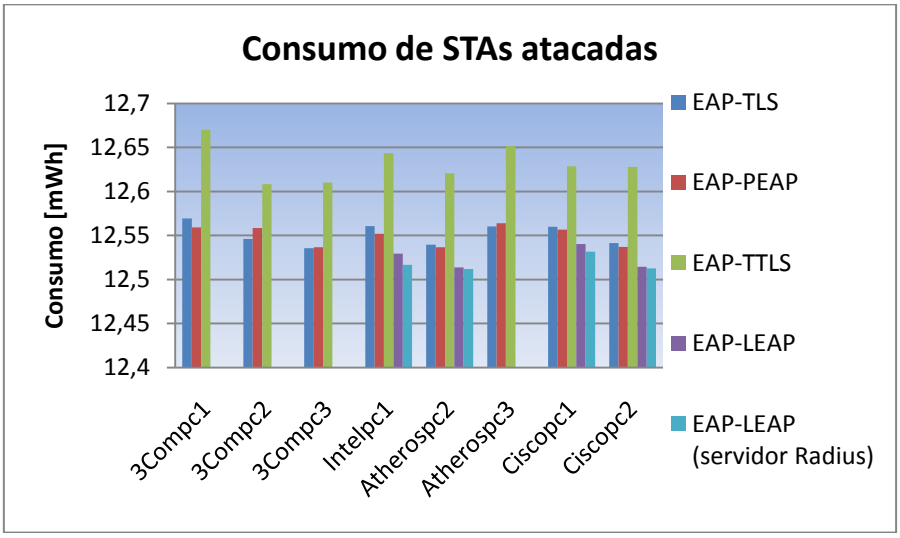


Fig J.6 Consumo STAs atacadas según el escenario.

J.2. Consumo de las STAs no atacadas

Para el cálculo del consumo de las estaciones no atacadas nos basaremos en el consumo del número de tramas que se puede enviar durante el tiempo de recuperación de un ataque (60s + Tiempo de autenticación). El consumo de las STAs no atacadas viene determinado por los diferentes tiempos y consumos durante los estados de inactividad, transmisión y recepción que se dan en una transmisión con éxito.

En los escenarios analizados, se toma como referencia el consumo para dos STAs.

- **EAP-TLS**

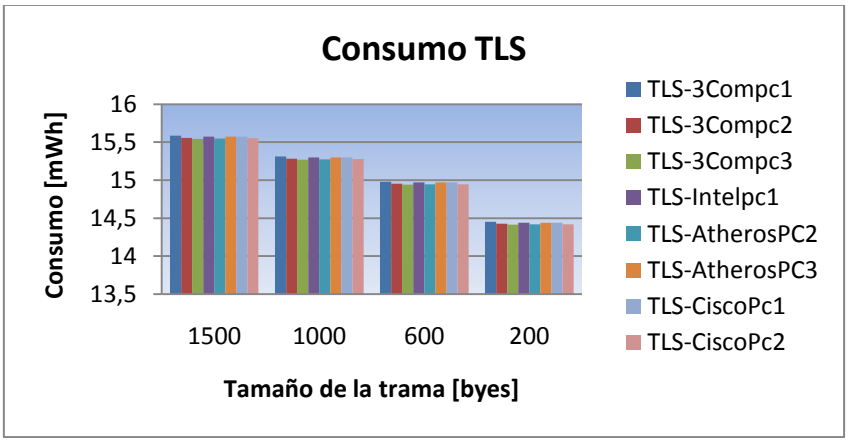


Fig J6. Consumo de las STAs no atacadas para EAP-TLS

- **EAP-PEAP**

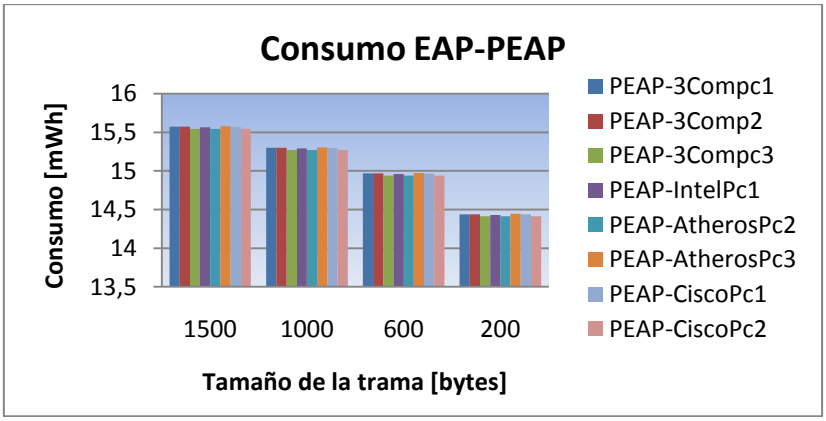


Fig. J7 Consumo de las STAs no atacadas para EAP-PEAP

- **EAP-TTLS**

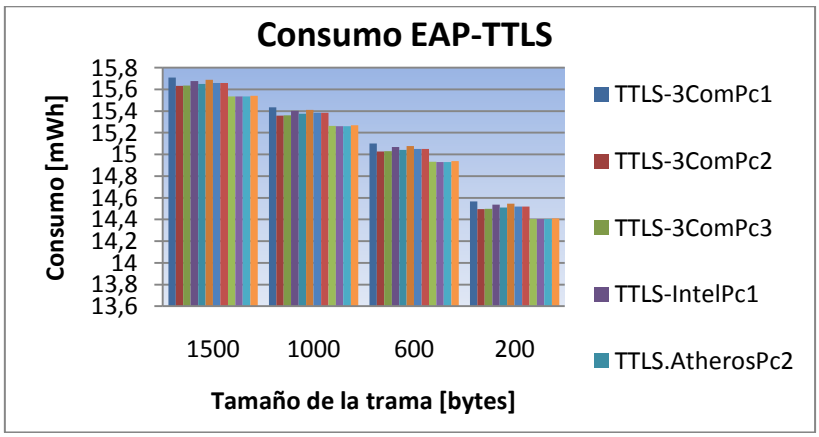


Fig. J8 Consumo de las las STAs no atacadas para EAP-TTLS.

- **EAP-LEAP**

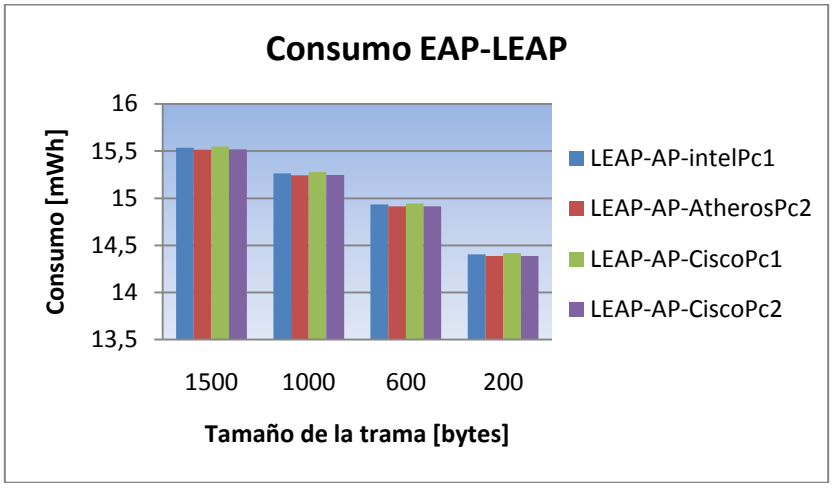


Fig. J9 Consumo de las STAs no atacadas para EAP-LEAP

- **EAP-LEAP (AP+ servidor Radius)**

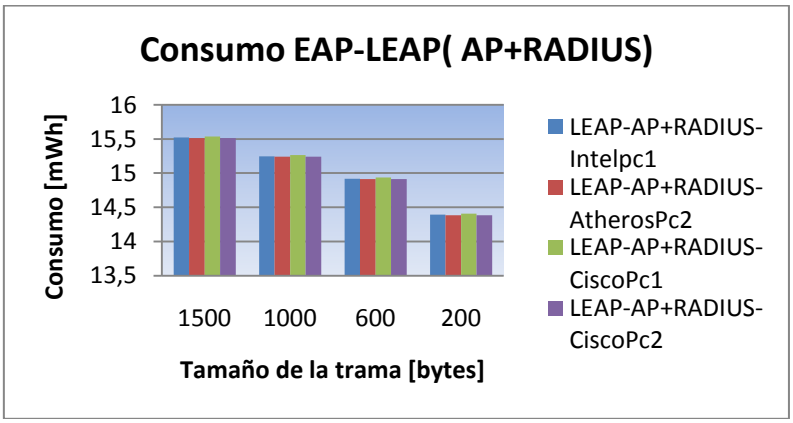


Fig. J10 Consumo de las STAs no atacadas para EAP-LEAP (servidor Radius)